

AD-A245 321



2

**AUTOMATED HANDLING AND ASSEMBLING OF NON-RIGID  
OBJECTS**

Fourth Semiannual Report

for period

16 July 1991 through 15 January 1992

for

**OFFICE OF NAVAL RESEARCH**

on

Contract No. N00014-90-J-1516

by

Yuan F. Zheng  
Principal Investigator  
Dept. of Electrical Engineering  
The Ohio State University  
Columbus, OH 43210

**DTIC**  
**ELECTE**  
**FEB 03 1992**  
**S D D**

This document has been approved  
for public release and sale; its  
distribution is unlimited.

92 1 29 015

92-02332

## TABLE OF CONTENTS

1. INTRODUCTION	1
2. NON-CONTACT 3-D DIGITIZING SYSTEM WITH SIX DEGREES OF FREEDOM	2
3. ROBOTIC HAND/EYE CALIBRATION BY ACTIVE VIEWING OF A CHECKERBOARD PATTERN	4
4. VIRTUAL SPACE OF DEFORMABLE OBJECTS HANDLED BY MANIPULATORS	6
5. CURRENT RESEARCH ACTIVITIES	8
APPENDIX A	
NON-CONTACT 3-D DIGITIZING SYSTEM WITH SIX DEGREES OF FREEDOM	10
APPENDIX B	
A NEW ROBOTIC HAND/EYE CALIBRATION METHOD BY ACTIVE VIEWING OF A CHECKERBOARD PATTERN	26
APPENDIX C	
VIRTUAL SPACE OF DEFORMABLE OBJECTS HANDLED BY ROBOT END-EFFECTORS	49



Accession For	
NTIS CRA21	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By <b>per A 231246</b>	
Distribution/	
Availability Codes	
Dist	Availability for
<b>A-1</b>	specification

## 1. INTRODUCTION

This is the fourth semi-annual report of the project entitled "Automated Handling and Assembling of Non-Rigid Objects". This report covers the period from July 16, 1991 to January 15, 1992.

The objective of this research project is to identify mechanisms for automated handling and assembling of non-rigid objects. To accomplish the goal, four aspects of the problem have been identified in our original proposal. They are:

- a. Study the deformation characteristics of non-rigid materials.
- b. Investigate optimal tool structure for handling non-rigid objects.
- c. Develop the sensing mechanism to locate non-rigid objects.
- d. Develop the motion control mechanism for the host machines.

To study the above four aspects, many technical issues need to be studied. In the past six months, three topics were studied by the Principal Investigator and his team. The first topic investigates the method for measuring a deformed surface using laser displacement sensors installed on articulated coordinate measuring machines (CMM). To precisely measure the profile of a unknown surface, touch probe has customarily been used to digitize the surface. The process of digitizing is very slow due to a large number of contacts that need to be made by the probe. We propose to use non-contact laser displacement sensor (LDS) to speed up the measurement. In addition, a multiple beam approach and an extrapolation approach are proposed to achieve high precision of measurement.

The second topic addresses a new method hand/eye calibration. This calibration problem arises when an active sensing mechanism is used for deformation identification. For active sensing, a vision camera is installed on a robot manipulator, since the latter can automatically position and orient itself in a large workspace. To guarantee precise measurement, the relative position and orientation between the camera and the manipulator must be calibrated. Existing methods of calibration are complicated and time consuming, and need expensive measuring equipments. We propose a new method in which the camera actively examines a checkerboard pattern for dimension calibration, and no expensive measuring equipments are needed. Experimental results prove that this method is simple while still maintaining a high precision.

The third topic studies the virtual space problem of deformed objects. The purpose is to identify a space which can cover all the possible shapes of a flexible object. By using the virtual space, a deformable object can be treated as a rigid object with the virtual space as its geometric shape. As a result, the difficult problem of path planning for deformable objects can be avoided.

In the rest of this report, we will summarize the results of the above three research activities.

## 2. NON-CONTACT 3-D DIGITIZING SYSTEM WITH SIX DEGREES OF FREEDOM

When a two-dimensional object is deformed, it becomes three-dimensional. Deformation identification amounts to find the three-dimensional shape of the deformed object. In recent years, computer controlled digitizing systems such as coordinated measuring machines (CMMs) become very popular to measure complex sculptured surface. Tradition digitizing systems use touch probe to measure individual points of the surface, which is a slow process since the probe has to make contact with the surface many times. We propose to use the LDS to replace the touch probe. A LDS can find the distance of a point without any contact using a triangulation principle. Without contacts, the speed of measurement can be greatly improved.

However, the LDS has certain limitations. First, the effective distance of a LDS is limited, and secondly, the orientation of the laser beam must be perpendicular to the object surface. If these two limitations are violated, either the LDS fails to give a measurement or the measurement is not precise. Since the surface is unknown, it is difficult to keep the LDS from violating the two limitations. We have developed an active sensing concept. The idea is to install LDS on an articulate CMM (a robot for example) such that the position and orientation of the LDS can be actively adjusted. For guiding the robot, we have developed two methods to predict the variation of the surface. They are the Multiple Beam Approach and the Extrapolation Approach.

The concept of the multiple beam approach can be explained by the illustration of Fig. 1. Consider a curve being probed by two light beams. The beams are separated by a short distance of  $b$ . Assume that a triangulation technique is applied to each beam. The distances from the surface to the light sources can be measured, and be denoted as  $L_1$  and  $L_2$  respectively. From Fig. 1, it can be seen that the tangential angle of the curve at the point that the first beam strikes the curve can be measured by

$$\theta = \arctg[(L_2-L_1)/b] \quad (1)$$

The angle calculated from (1) is the orientation that needs to be altered by both beams such that the beams can be perpendicular to the beam. The difference between  $L_1$  and  $L_2$  are used to adjust the range of the LDS such that they are kept from getting away from the object surface. It should be noted that only the measurement made by the first LDS is used as the real measurement.

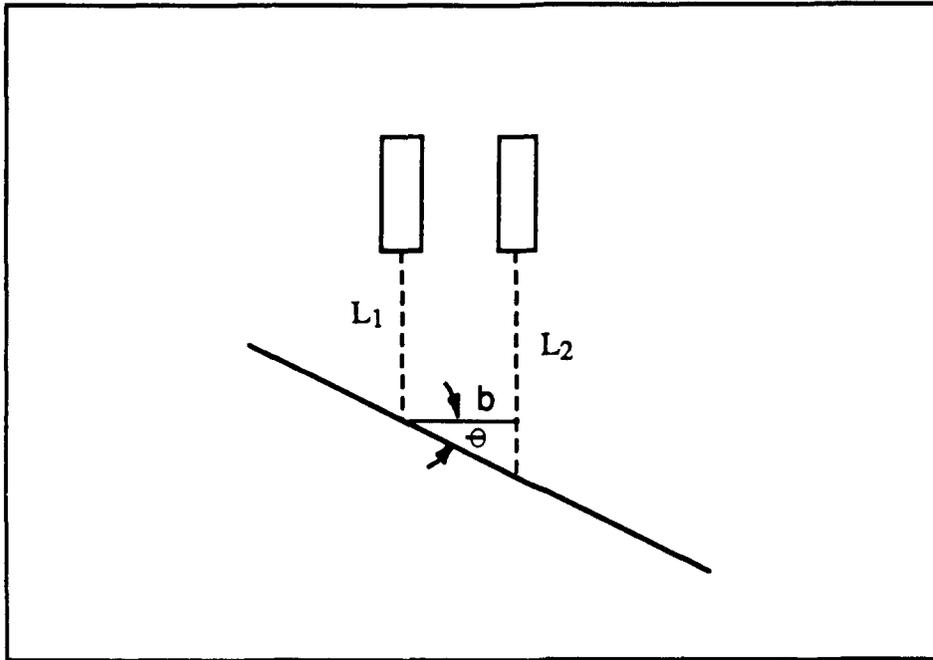


Fig. 1 The Multiple Beam Approach

The above approach for surface following only uses two measuring points for surface prediction. If the surface has a large variation, one can use a general  $n^{\text{th}}$  order polynomial extrapolator approach to extrapolate the surface.

The  $n^{\text{th}}$  order polynomial extrapolator is the process of extrapolating a continuous function between the present sampling instants and the next sampling instant such that the next sampling instant can be predicted. The polynomial has the following standard form:

$$z(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 \quad (2)$$

The  $n+1$  unknown coefficients in the polynomial are solved for by using the previous  $n+1$  data points. For example, if a 2nd Order Polynomial Extrapolator is used, there are 3 unknown coefficients and 3 measured data points are needed. If the data points in the  $z$  direction are measured at points  $0$ ,  $-x_s$  and  $-2x_s$  along the  $x$ -axis, the measured data points can be written in the following form:

$$\begin{aligned} z(0) &= a_2(0)^2 + a_1(0) + a_0 \\ z(-x_s) &= a_2(-x_s)^2 + a_1(-x_s) + a_0 \\ z(-2x_s) &= a_2(-2x_s)^2 + a_1(-2x_s) + a_0 \end{aligned} \quad (3)$$

It is clear that the coefficients of  $a_0$ ,  $a_1$  and  $a_2$  can be solved by using the measurements on  $z(0)$ ,  $z(-x_s)$  and  $z(-2x_s)$ . Once the coefficients are solved,  $z(x_s)$  can be predicted by using the same polynomial.

The approach presented in this subsection has been tested in our laboratory using two Keyence LB-72 LDS. The two LDS are installed on the hand of a PUMA robot which is to function as an articulated CMM. Successful results of the experiment prove that the proposed approach is valid.

The details of this research activity has been written as a technical paper, which is attached with this report as Appendix A.

### **3. ROBOTIC HAND/EYE CALIBRATION BY ACTIVE VIEWING OF A CHECKERBOARD PATTERN**

The LDS approach as described above is used to detect details of a deformed shape which is called local deformation. For identifying global deformation, computer vision system with CCD cameras needs to be used. In order to completely identify the shape of a deformed object, the cameras have to be placed in different positions and with different orientations, which is called active sensing. Eye-on-hand configuration is very effective for this purpose. That is, a camera is installed on a robot end-effector. Since the scene through the camera is only relative to the camera, the relative position and orientation between the camera and the robot hand must be known. This is called the hand/eye calibration problem.

Previous calibration methods need to solve a large set of nonlinear equations which describe the relations between the coordinate systems of the camera and the end-effector. In addition, a fixed target needs be installed in the robot workspace. The target serves as an intermediate stage in the hand/eye calibration, and its position and orientation in the robot coordinate system must be precisely known by using expensive measuring equipments. By processing the image of the target, the position and orientation of the target in the camera coordinate system are determined. By solving the nonlinear equations relating the position in the world and camera coordinate systems, the calibration procedure can be completed.

This procedure needs expensive equipments to precisely determine the position and orientation of the fixed target in the robot coordinate system. Furthermore, the camera itself must first be calibrated. That is, the relation between the position of a point in the camera coordinate system and its position in the image must be precisely known.

Two kinds of camera model are commonly used for camera calibration, the pinhole model and the two-plane model. The disadvantages of the pinhole method are the complexity of the implementation and the requirement of prior knowledge of camera parameters such as the focal

length, scale factors and lens distortion parameters. The two-plane method, on the other hand, is much simpler to use. It does not require to derive any specific camera parameters. It only needs to establish correspondence between a set of calibration points on two parallel target blocks and their project points in the image frame. Once the correspondence is established, the position of a point in the camera coordinate system can be determined from its position in the image frame.

The two-plane model cannot be directly used to calibrate the hand/eye relation. We propose a new method which adapts the two-plane model, and extends the model to calibrate the hand and eye relation. Thus, precise measurements of the target block become unnecessary. The method has the following two features:

- a. The optical axis of the camera in the coordinate space of the hand coordinates is first found. By using the optical axis, the position and orientation of target block can be found by the camera. As a result, one does not need expensive equipments to determine the position of the target block in the robot coordinate system.
- b. A 6 by 6 evenly spaced checkerboard (Fig. 2) is used as target block. Each square on the checkerboard is colored with one of the three different gray levels. By using this specially designed target block, it is easy and fast to process the image and to find the calibrated points on the target block.

We have tested the new method in our laboratory. The results of the experiment have shown that this new method, without the use of expensive equipments, results in an accurate calibration between the camera and the robot hand.

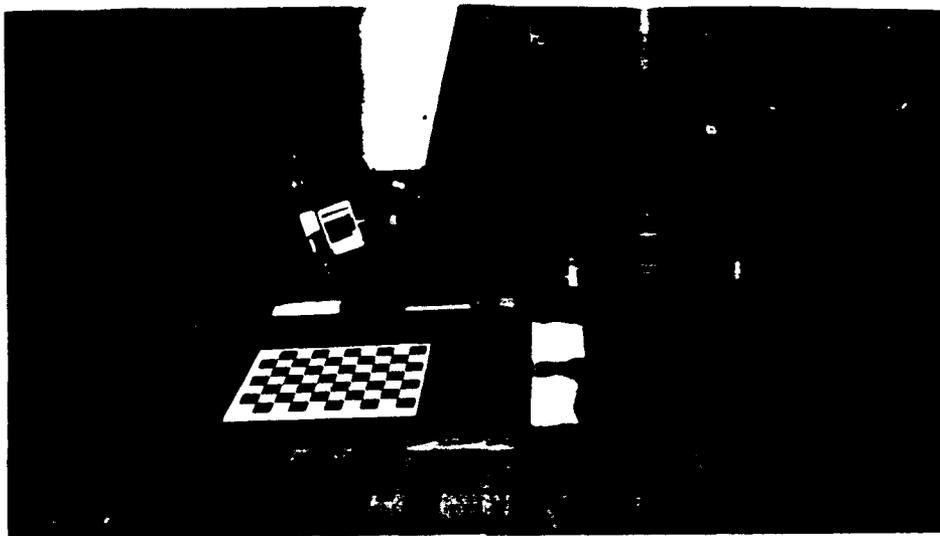


Fig. 2 Hand/Eye Calibration Using a Checkerboard Pattern

The detail description of this new hand/eye calibration method is presented in the paper entitled "A New Robotic Hand/Eye Calibration Method by Active Viewing of a Checkerboard Pattern", which is attached as Appendix B of this report.

#### 4. VIRTUAL SPACE OF DEFORMABLE OBJECTS HANDLED BY MANIPULATORS

When a deformable object is moved by a machine, the shape of the object is uncertain because of deformation. As a result, it is difficult to plan a collision-free path when the object is to be moved in an obstacle-filled environment.

We propose to use a virtual space approach. The virtual space is a space which covers all the surface and edges of a deformable object (Fig. 3). By using the virtual space, the deformable object can be treated as a rigid body with the virtual space as its geometric shape. As a result, the motion of the object can be programmed using existing techniques for rigid bodies. Consequently, the uncertainty problem of the object shape is avoided.

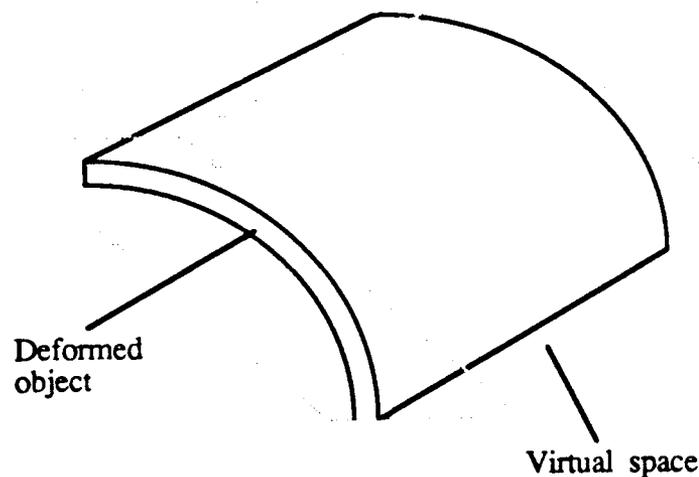


Fig. 3 Virtual Space of a Deformable Object

To find the virtual space of a deformable object, deformation behavior of the object must be known. Our study has been concentrated on one and two-dimensional objects since these objects are most frequently used in the manufacturing process, such as the production of printed-wiring-boards in the electronics industry and composite materials in the aerospace industry.

When a deformable object is grasped by the end-effector of a manipulator, two types of deformation are possible. In the first case, deformation occurs when the end-effector pitches (Fig. 4a). In the second case, deformation occurs when the end-effector rolls (Fig. 4b). The first step of our study is to find the deformation behavior of the object in the above two cases.

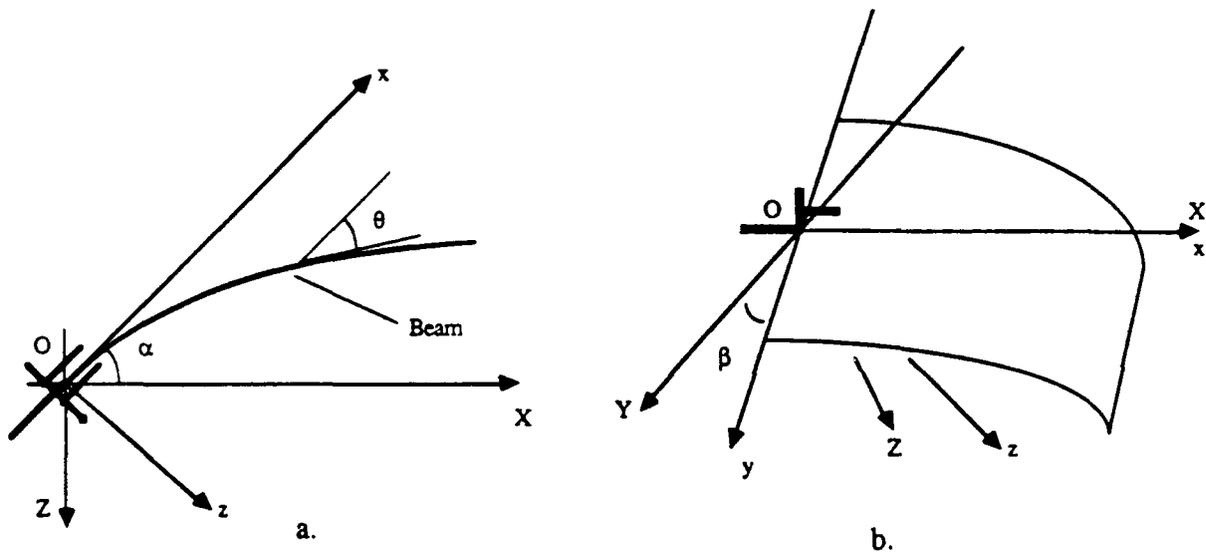


Fig. 4 The pitch and roll deformation

To find the deformation behavior, the governing equation of the deformations is first derived from the classical mechanics. In the case of pitch deformation, the equation can be written as:

$$\frac{d^2}{dx^2} (EI \frac{d^2 w}{dx^2}) = v(x) \quad (4)$$

where  $x$  represents the axis of deformation,  $E$  is the modulus of elasticity,  $I$  is the area moment of inertia,  $v(x)$  is the load of the beam and  $w$  denotes the deformation. There does not exist a closed form solution for the highly nonlinear equation (4). We use a perturbation technique to find an approximate solution. The solution is expressed as an ascending perturbation series of  $\epsilon$  which is a small parameter:

$$w(x) = \sum_{n=0}^{\infty} w_n(x) \varepsilon^n \quad (5)$$

By substituting (5) into (4), one can obtain the solution of (4) to be

$$w(x) = \left( \frac{1}{24} x^4 - \frac{1}{6} a x^3 + \frac{1}{4} a^2 x^2 \right) \frac{p}{EI} \cos \alpha + \frac{1}{1680} \left( (a-x)^7 + 7 a^6 x - a^7 \right) \left( \frac{p}{EI} \right)^2 \sin \alpha \cos \alpha + \dots \quad (6)$$

where  $p$  is the load per unit length of the plate, and  $\alpha$  is the pitching angle of the end-effector. By using the solution of (6), one can identify the maximum deformation of the object. From the maximum deformation, a virtual space can be determined for a deformable object. The deformation solution in the rolling case can be likewise solved.

While determining the virtual space, we propose to use the following two criteria for selecting an optimal virtual space:

- Criterion 1:** The virtual space should have as a small volume as possible while covering all the possible deformed shapes.
- Criterion 2:** The computation for determining the virtual space should be as simple as possible.

The first criterion is for making effective use of the robot workspace. The reason for the second criterion is self-explanatory. In reality, the two criteria are often trade-offs of each other. That is, a smaller virtual space needs more computation to determine.

The detailed description of the results of this research has been written as a technical paper which is attached as Appendix C of this report.

## 5. CURRENT RESEARCH ACTIVITIES

Currently, we are engaged in the following research activities.

- a. With the calibration problem being solved, we are investigating strategies for using the active sensing to identify deformation. One approach is to integrate active and passive sensing. The goal is to obtain complete 3-D information by a minimal motion of the manipulator.
- b. Once the shape of a deformed shape is measured, the parameters which governs the deformation behavior of the object can be calculated. These parameters can be used to

predict the shape of deformation when the external conditions change. As a result, one does not need to constantly measure the deformed shape by vision sensors. At this point, we are investigating the algorithms for calculating the parameters using visual images. A Finite Element Method (FEM) is being used in the research.

- b. The third problem currently being studied is the precision issue of the non-contact digitizing system. The digitizing approach described in this report uses an articulated CMM. The kinematics of an articulated structure is nonlinear. We are investigating optimal configurations in which an articulated CMM is most precise, and approaches for maintaining such configurations.

The results of the above research activities will be presented in the next semi-annual report.

## APPENDIX A

### NON-CONTACT 3-D DIGITIZING SYSTEM WITH SIX DEGREES OF FREEDOM

Kevin B. Smith and Yuan F. Zheng  
Dept. of Electrical Engineering  
The Ohio State University  
Columbus, OH 43210

#### ABSTRACT

This research investigates digitizing methodologies for deformed surface identification using laser displacement sensors (LDS). From studying the existing technology, maintaining a near normal sensor-to-surface orientation was found to be key issue in obtaining a precise reconstruction of deformed surface. Since the surface is unknown, however, it is difficult to predict such an orientation. A new methodology is proposed to achieve a precise measurement. The new methodology incorporates a six degrees-of-freedom robot and laser displacement sensors (LDS). Variations in surface contour are locally predicted by using multiple LDS or an extrapolation approach. Using this result, the ideal sensor-to-surface orientation can be maintained. This paper describes the proposed sensor-to-surface orientation techniques and details the system implementation. Also, experimental verification of the proposed techniques is presented.

---

ACKNOWLEDGEMENT: This work was supported by ONR under grant N00014-90-J-1516, and by an industrial fellowship from the Engineering Research Center of The Ohio State University.

## 1. INTRODUCTION

In recent years, digitizing unknown surfaces has become increasingly important to the manufacturing industry. For example, in the automobile industry, draw die engineers often build a ceramic draw die model in order to design the draw die addendum surfaces. The addendum surfaces are constructed around the peripheral of the model's product surfaces (e. g. an automobile body part). Due to their complexity, the addendum surfaces are created by hand on the ceramic models [1]. The engineer uses his or her own experience to determine the "correct" surfaces to use. Once designed, these surfaces are digitized to create a data base that the computer-aided-design (CAD) software can use.

Another application of the digitizing technology is in "rapid part acquisition" or so called "3-D copying" [2]. This is a process which reduces the time and cost to manufacturing critical parts whose CAD model is not available. Such a process is important in the efficient and effective maintenance of valuable equipment utilized in both military and civilian organizations.

In our research of deformation identification, it is found that computer vision systems cannot identify details of local deformation. A digitizing process becomes necessary, when the deformed shape of a non-rigid object needs to be measured.

Although the digitizing technology is increasingly important in today's manufacturing activities, many aspects of the technology still need to be enhanced. As illustrated, the digitizing process needs to be fast, accurate, reliable, and nondestructive. In the past, coordinate measuring machines (CMM) using touch probes have been used for digitizing surfaces. Digitizing using a CMM with a touch probe is accurate and reliable; however, it is also slow. This has motivated several tool manufactures to replace the touch probe with a Laser Displacement Sensor (LDS). In addition, 3-axis machining centers are now being used to position the LDS while digitizing.

The LDS can collect surface data faster than a touch probe. In addition, the LDS does not require mechanical parts to come in contact with the surface, thus protecting the surface from possible damage or from being further deformed. However, the LDS's accuracy depends on the position and orientation of the laser with respect to the surface. The LDS is most accurate when its orientation is nearly perpendicular to the surface and the surface is within the LDS's effective stand-off range.

On 3-axis machines, the LDS is held in a fixed orientation while digitizing. This limits the types of surfaces that can be digitized. Because the LDS's orientation must be within a specified range of sensor-to-surface perpendicularity, only surfaces that fall within that range can be digitized. Work has been done to increase the LDS's surface-to-sensor orientation range [3]. However, by using a six degree-of-freedom robot for positioning and orienting the LDS, a more

general surface (one with high tangential variations) can be digitized. Use of an articulated machine is suggested by the authors. The LDS's position and orientation can then be continuously changed during the digitizing process. Thus, narrower variations in sensor-to-surface orientation can be maintained.

To study the issues involved with maintaining a near perpendicular orientation between the surface and LDS, a non-contact 3-D digitizing system with six degrees-of-freedom was constructed. The system uses two LDS's which are positioned using a robot manipulator. Two different approaches to maintaining sensor-to-surface orientation while digitizing were investigated, namely the Multiple Beam Approach and the Extrapolation Approach.

In the Multiple Beam Approach, multiple laser displacement sensors are used to detect the tangential angle and the curvature of the surface. Based on the tangential angle and the curvature, the orientation of the LDS can be adjusted until it is perpendicular to the surface. In the Extrapolation Approach, an  $n^{\text{th}}$  order polynomial is used to extrapolate the surface's height at the next digitizing location. To obtain the complete description of the polynomial,  $(n+1)$  unknown coefficients of the polynomial must be solved. This is done through previous  $(n+1)$  measurements of the surface.

The structure of the paper is as follows. Section 2 outlines the mathematical background for the two proposed digitizing approaches. Section 3 describes the hardware and software used to construct the digitizing system. Section 4 presents the experimental results of the proposed digitizing approaches. Finally the paper is concluded with a summary.

## 2. MATHEMATICAL BACKGROUND

### A. Multiple Beam Approach

The concept of the Multiple Beam Approach can be explained by the illustration of Fig. 1. Consider a curve being probed by two laser displacement sensors. The laser beams of the LDS are separated by a short distance of  $b$  (the selection of the length of  $b$  will be discussed later). The distances from the surface to the light sources can be measured by the LDS, and be denoted as  $L_1$  and  $L_2$  respectively. From Fig. 1, it can be seen that the tangential angle of the curve at the point that the first beam strikes the curve can be measured by

$$\theta = \arctg[(L_2-L_1)/b] \quad (1)$$

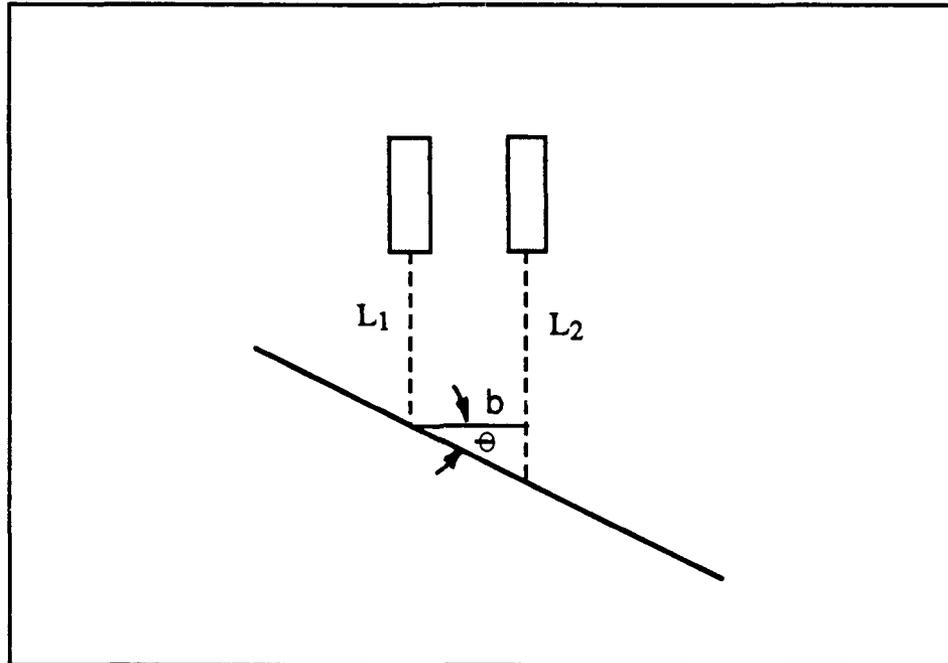


Fig. 1 Multiple Beam Approach

The angle calculated from (1) is the orientation that needs to be altered by both beams. The difference between  $L_1$  and  $L_2$  can be used to adjust the range of the LDS such that the first LDS is kept from moving away from the object surface. It should be noted that only the measurement made by the first LDS will be used as the surface measurement.

The distance between the two light beams,  $b$ , is determined by the gradient of the surface. If the gradient changes rapidly on a surface,  $b$  must be very small. Otherwise, the scanner may not be able to follow the surface. Ideally,  $b$  is adjustable to accommodate various surfaces and precision requirements.

If three laser beams are used, the radius of curvature of the surface can also be determined, which can be approximately expressed as

$$\rho = \frac{b}{\Delta\theta} \quad (2)$$

With the radius of curvature being identified, the LDS can predict the tangential angle of the surface at the next digitizing points. The adjustment can be made while the LDS is being moved to the next digitizing points. It can be seen that as more laser displacement sensors are used, the future variation of the surface is better known, and the adjustment of the LDS can be more efficiently made to maintain the perpendicular orientation. The cost for both hardware and computation, however, becomes higher as the number of LDS increases.

## B. Extrapolation Approach

The extrapolation approach uses an  $n^{\text{th}}$  order polynomial to extrapolate the surface at the next digitizing location. The polynomial has the following standard form:

$$z(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 \quad (3)$$

The  $(n+1)$  unknown coefficients in the polynomial are solved for by using the previous  $(n+1)$  measurements of the surface. For example, a  $2^{\text{nd}}$  order polynomial uses the last three measured surface points to solve for the unknown coefficients  $a_2$ ,  $a_1$ , and  $a_0$ . Let  $[(x_1, y_1, z_1)$ ,  $(x_2, y_2, z_2)$ , and  $(x_3, y_3, z_3)]$  represent the three most recently measured points, see (Fig. 2).

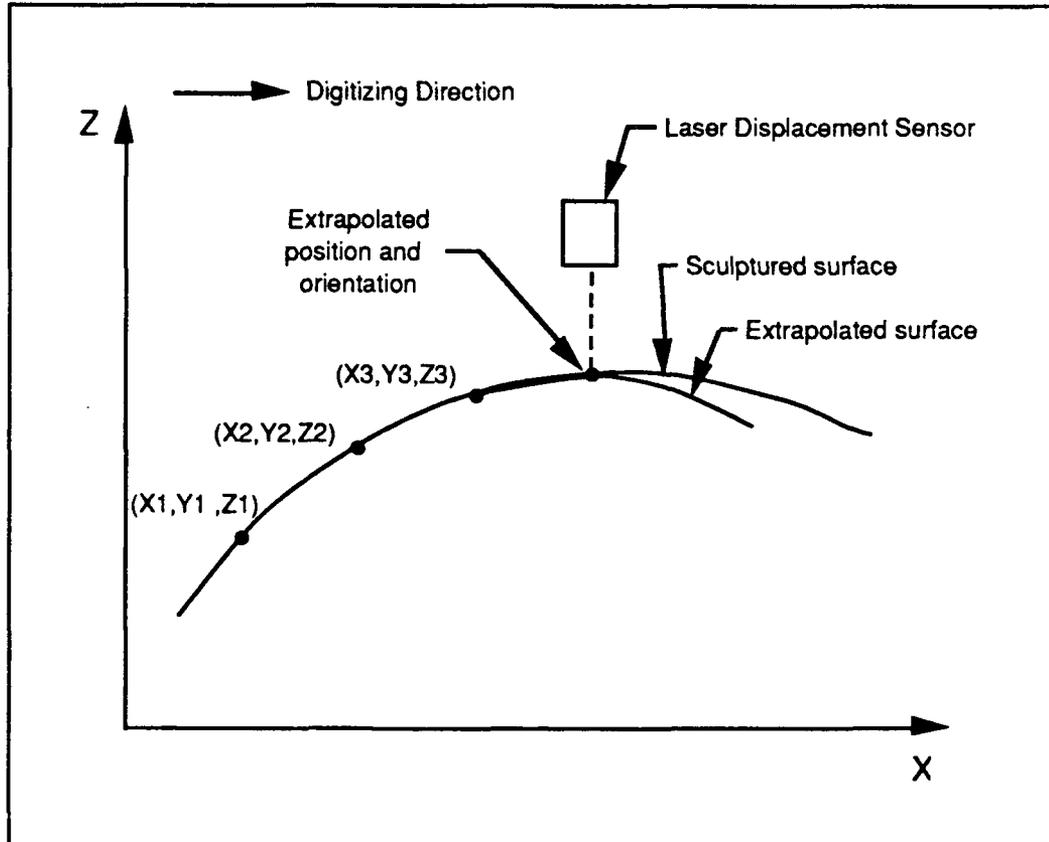


Fig. 2 2nd Order Polynomial Extrapolation

If digitizing in the  $x$  axis direction, then  $y_1$ ,  $y_2$ , and  $y_3$  are all equal. In addition, each point must satisfy equation (3). Therefore, from equation (3), the three constraining equations are:

$$z_1 = a_2x_1^2 + a_1x_1 + a_0$$

$$z_2 = a_2x_2^2 + a_1x_2 + a_0$$

$$z_3 = a_2x_3^2 + a_1x_3 + a_0$$

Once the coefficients are known, equation (3) can be used to extrapolate the surface's height at the next location to be digitized. Surface orientation can also be extrapolated by evaluating the derivative of equation (3), see equation (4).

$$z'(x) = (n)a_nx^{n-1} + (n-1)a_{n-1}x^{n-2} + \dots + a_1 \quad (4)$$

Equation (4) gives an extrapolated tangent value of the surface. The orientation of the LDS is set perpendicular to this tangent value. Thus, the laser beam is oriented so that its slope is set equal to the negative inverse of  $z'$ . The  $z$  axis position for the LDS is calculated by equation (3). The  $x$  axis value for the position is used in equations (3) and (4) and is a fixed distance apart from the last measured point. This distance is selected by the operator at run time.

Unfortunately, this extrapolation approach needs " $n+1$ " surface points before it can be implemented. For  $n=2$ , three surface points must initially be measured using a different approach before implementing the 2<sup>nd</sup> order polynomial extrapolation approach. One solution is to use the polynomial extrapolation approach but limit the extrapolation order to the highest order that fits the data collected. Therefore, for  $n=2$ , a 0<sup>th</sup> order polynomial is used to extrapolate the 2<sup>nd</sup> point, a 1<sup>st</sup> order polynomial is used to extrapolate the 3<sup>rd</sup> point, and a 2<sup>nd</sup> order polynomial is used to extrapolate the remaining points. Note, a 0<sup>th</sup> order polynomial extrapolates the surface as a plane with a constant  $z$  axis value.

A variation to the approach just described is to use the robots initial orientation when extrapolating the 2<sup>nd</sup> point. In this case, the surface is still assumed to be a plane but the plane is assumed to be perpendicular to the initial orientation of the robot. This is a better assumption because the operator would naturally place the robot in a near perpendicular orientation before starting the digitizing process. This variational approach was used in the laboratory to extrapolate the 2<sup>nd</sup> point. The 3<sup>rd</sup> point was extrapolated using a 1<sup>st</sup> order polynomial. And subsequent points were extrapolated using a 2<sup>nd</sup> order polynomial.

Another digitizing location that needs special consideration, is the 1<sup>st</sup> point on subsequent rows. When the surface has been digitized across the requested  $x$  length, the LDS is moved in the  $y$  direction and the digitizing resumes in the opposite  $x$  direction. The  $x$  value,  $z$  value and orientation for the first point of the new row is assumed to be the same as the last point of the preceding row, only the  $y$  axis value is different. Subsequent points on the new row are extrapolated as they were at the start of the preceding row. Namely, the 2<sup>nd</sup> point on the row is

extrapolated assuming the surface is a plane perpendicular to the LDS. The 3<sup>rd</sup> point in the row is linearly extrapolated by using the 1<sup>st</sup> and 2<sup>nd</sup> points of the row. All subsequent points in the row are extrapolated using a 2<sup>nd</sup> order polynomial.

The two approaches as defined in subsections **A** and **B** can also be combined. If multiple beams approach is used, it will be relatively easy to define the position and orientation of the LDS in measuring the first few digitizing points. That is, one can use the multiple beam approach as defined in **A** to measure the first few points. Once enough points are measured for the polynomial, the extrapolation approach can take over. The advantage of the extrapolation approach is that one can predict the variation of the surface by making one measurement at a time.

### 3. SYSTEM SETUP

#### A. Hardware

The system consists of two Laser Displacement Sensors, a six degrees-of-freedom PUMA robot, and an IBM PC (see Fig. 3a and Fig. 3b).

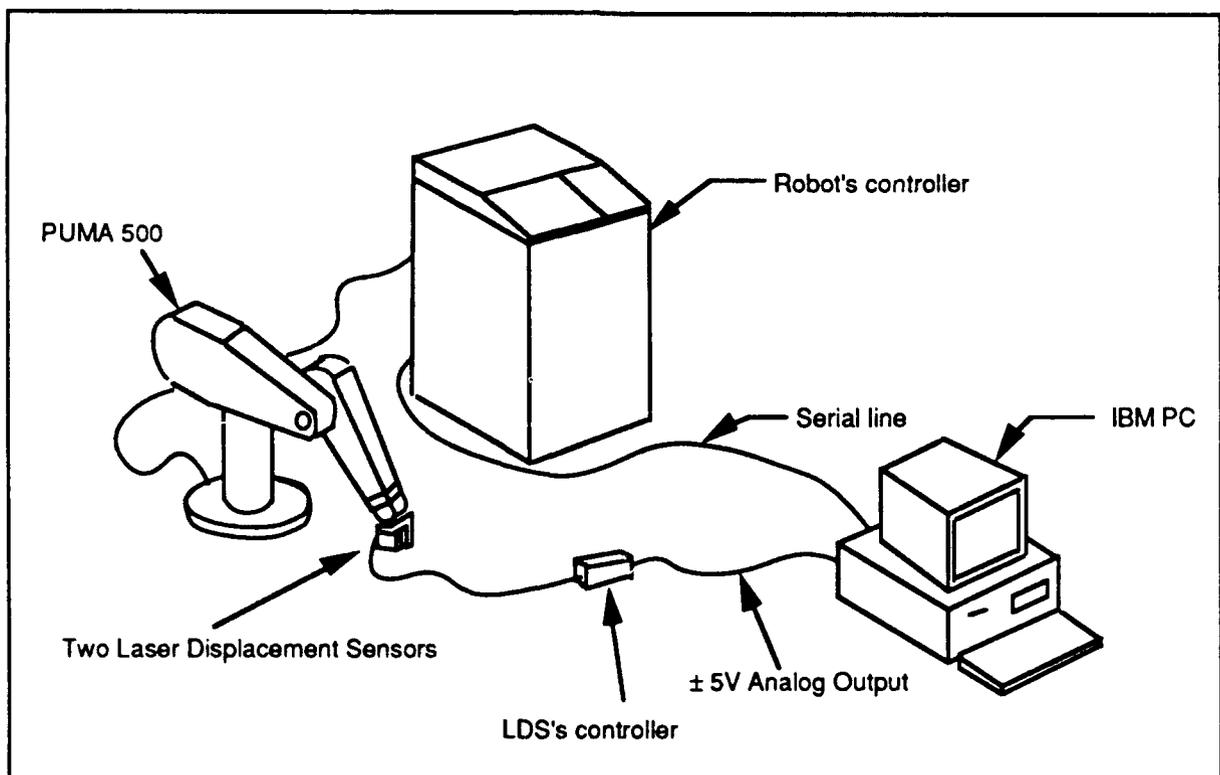


Fig. 3a Major Components in Digitizing System

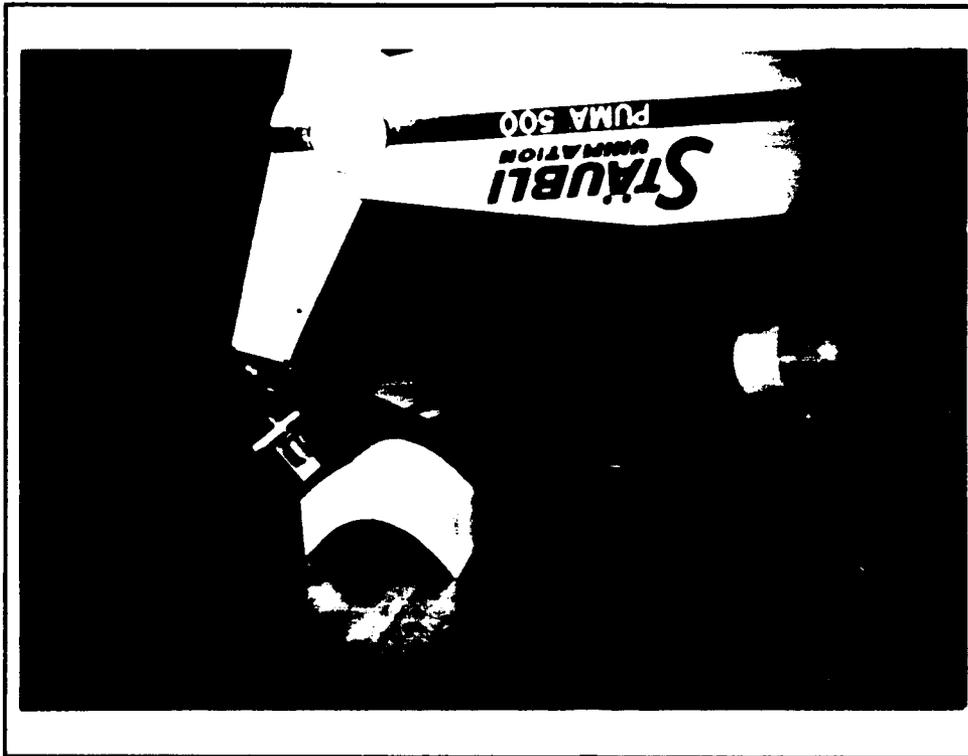


Fig. 3b Picture of digitizing system

A fixture mounted on the end-effector positions the two LDS' such that the laser beams are parallel to each other and separated by a distance of 17.85 mm. The analog  $\pm 5$  volt output of the two LDS is converted to a digital value by a MetraByte DAS-8 board in the IBM PC. The IBM PC communicates to the PUMA robot's controller via a serial port. As needed, the control software on the IBM PC requests the robot's current position and directs the robot to move. The control software uses the LDS data and the robot's position-orientation information to determine data points on the surface being digitized. These data points are then used to implement the multiple beam approach or the extrapolation approach described above.

Selected specifications on the major hardware components are listed in Table 1.

## B. Software

The control software was made up of several low level and high level modules. The low level modules include: Serial communication, Screen memory management, and data acquisition. The high level modules include: protocol to communicate with robot's controller, terminal emulation mode, digitizing mode, and a stand alone user program running on the robot's controller, see (Fig. 4).

- Keyence LB-72/LB-12: Laser Displacement Sensor
  - Stand-off distance: 40 mm
  - Measurement range:  $\pm 10$  mm
  - Resolution:  $\pm 0.015$  mm (Object: white paper at a distance of 40mm, at 2ms)
  - Linearity: 1% of the displacement (Object: white mat paper, tilted  $\pm 30^\circ$ )
- PUMA Mark III 500 Series Robot
  - 6 axes of rotation,
  - 914 mm (3.0 ft) reach range,
  - Position Repeatability  $\pm 0.1$  mm ( $\pm 0.004$  in)
- IBM PC
  - 4.77 MHz clock speed
- MetraByte DAS-8: Data Acquisition Board
  - $\pm 5$  volt, 12 bit Analog to Digital Converter

Table 1 Selected Hardware specifications

The low level serial communication software was constrained to be interrupt driven due to the speed at which the robot's controller transmitted serial data. The interrupt service routine, IRQ4, written for this experiment, uses two circular queues to buffer in-coming and out-going data. When called, this routine checks the 8250 UART for receive-data ready. If there is in-coming data, the character is placed in the in-coming queue. Otherwise, if the transmitter holding register is empty then a character for the output queue is transmitted. The interrupt service routine executes fast enough to prevent an overflow from occurring. Thus, the high level control software only reads from and writes to the in-coming and out-going queues respectively to communicate with the robot's controller. When the control software wants to transmit data to the robot's controller, it fills the output queue with data and calls a flush routine which enables interrupts when the UART's transmitter-holding register is empty. After the out-going queue is empty, the flush routine disables interrupts on transmitter-holding-register empty and returns to the calling functions.

The screen memory management routines divide up the screen memory into two windows. The top window is used for displaying terminal emulation mode data. The bottom window is used for the digitizing control mode. Scrolling is supported in both windows. The center line of the screen memory is used for displaying current LDS readings.

The data acquisition module is used to read the LDS voltage values and convert them to displacement units (millimeters). The DAS-8 board uses 12 bits for the analog-to-digital conversion. Therefore, the displacement measurements are quantized in increments of 0.005 mm ( $20 \text{ mm} / 2^{12} = 5 \text{ mm}$ ). This quantization value is 5% of the magnitude of the robot's position repeatability and is therefore considered negligible. An out-of-range error signal from the LDS is

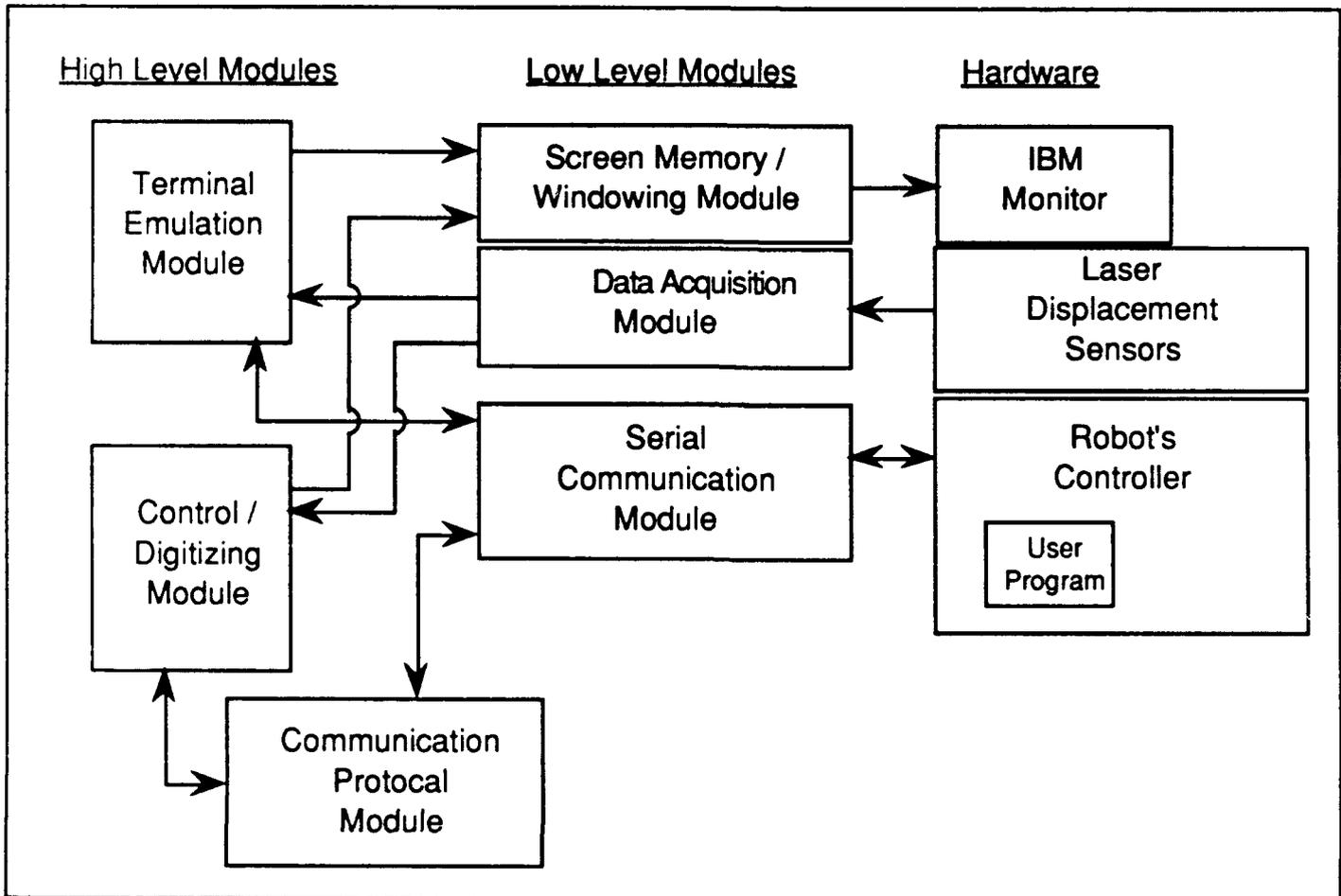


Fig. 4. Control Software Modules

also read and the results are passed on to higher level modules. If an out-of-range error is detected, then the higher control modules try reorienting the LDS in order to get a good reading.

Higher level functions are written to handle the protocol of the robot's controller. These functions oversee the format of data transmitted to the robot's controller. Because the robot's controller operates in full-duplex, all data transmitted to the robot is checked for transmission errors. If a transmission error is detected, an abort signal (^U) is sent to the robot's controller and the command line is re-transmitted.

This software first runs in terminal emulation mode. In this mode, the robot is operated as if it is connected to a terminal and the robot can be manually positioned into its starting position before digitizing begins.

The control mode modules coordinates and controls the digitizing operation. This module reads the LDS values and the robot's position and orientations. It implements the surface-to-sensor orientation approaches and saves all digitized surface data points in a file. The control mode is

initiated from the terminal mode by using the hot key, (^K). Once initiated, the software first prompt the user for overall control information, such as: output data file name, size of area to digitize, number of digitizing points to collect, and orientation approach to use.

A small VAL-II user program is executed on the robot's controller concurrently with the control software. This program is menu driven and has two options. Option 1 outputs the robot's current position and option 2 reads (X,Y,Z,O,A,T) data to reposition the robot. The program uses prompts to identify which mode it is in. The "?" prompt signifies the program is waiting for an option selection. The program displays a ">" prompt when it is waiting for (X,Y,Z,O,A,T) data and it uses the "\*" prompt to signify that the robot has not completed a move.

## 4. EXPERIMENT

### A. Calibration

Before using the system, the Keyence LB-72/LB-12 Laser Displacement Sensors were calibrated and a transformation matrix from the end-effector to the LDS was defined. The LDS have two calibration adjustments. One is the 0-ADJ adjuster trimmer which adjusts the 0 volt stand-off distance. The stand-off distance on one LDS was set to roughly the center of its measuring range. Then, the other LDS was adjusted carefully to have the same stand-off distance as the first LDS. The second calibration adjustment is the SPAN adjuster trimmer. This adjustment varies the volts per displacement value. Both LDS, were calibrated to 5 volts per 10 mm displacement.

After the lasers were calibrated, the transformation matrix from the end-effector to the 0 volt reference position was determined. The PUMA robot's controller has a reserved name for this transformation matrix, named "TOOL". Once this transformation matrix is defined, the robot can be positioned using locations for the LDS rather than the end-effector.

### B. Digitizing Rate

The overall digitizing rate was measured in the laboratory. The results are shown in Table 2. The rate was measured while using the Multiple Beam Approach to digitize a plane.

<u>Motor Speed</u>	<u>:</u>	<u>Sampling Distance</u>	<u>:</u>	<u>Digitizing Rate</u>
5%	-	10 mm	-	0.77 (data points / second)
5%	-	1 mm	-	0.91
100%	-	10 mm	-	1.00
100%	-	1 mm	-	0.91

Table 2 Digitizing Rate

The serial communication and error checking between the IBM PC and the robot's controller is believed to contribute significantly to the time needed to digitize. Speed improvements could possibly be made by using a more sophisticated communication protocol available on the robot's controller. The serial communication interface used was chosen because it was thought to be faster to implement.

### C. Surface-to-Sensor Orientation Techniques

Both the Multiple Beam Approach and the Extrapolation Approach to maintaining surface-to-sensor orientation were demonstrated by digitizing a simple concave and convex surface. The surface is shown in Fig. 3b and the digitized data is shown in Figs 5 and 6.

Regardless of the approach used, the surface is digitized in a grid pattern. The distance between adjacent points is the same in any row or in any column. The surface is digitized row by row, where a row is parallel to the x axis. When the end of the row is reached, digitizing continues on the next row but in the opposite direction. Therefore, if a row is digitized in the positive x direction, then the next row will be digitized in the negative x direction. For the Multiple Beam Approach, the second laser would be leading the digitizing laser on one row and following the digitizing laser on the next row.

The Multiple Beam Approach gave consistent, reliable results provided the surface's radius of curvature was large in comparison with the distance between the two lasers. Smaller radius of curvatures produce larger errors between the predicted and the actual position and orientation of the surface. This observation is, to some degree, intuitive if you consider digitizing the cross-section of a semi-circle which has a smaller diameter than the distance between the two laser beam. In this extreme case, only one laser beam, at most, shines on the semi-circle at any time. Under this condition, the Multiple Beam Approach fail. However, the Extrapolation Approach could be used provided the digitizing interval is small in comparison with the radius of curvature.

A 2nd order polynomial extrapolation method was implemented. This order of polynomial was observed to be sensitive to errors in surface point measurements. This sensitivity is increased with smaller digitizing intervals. This is illustrated in Fig. 7.

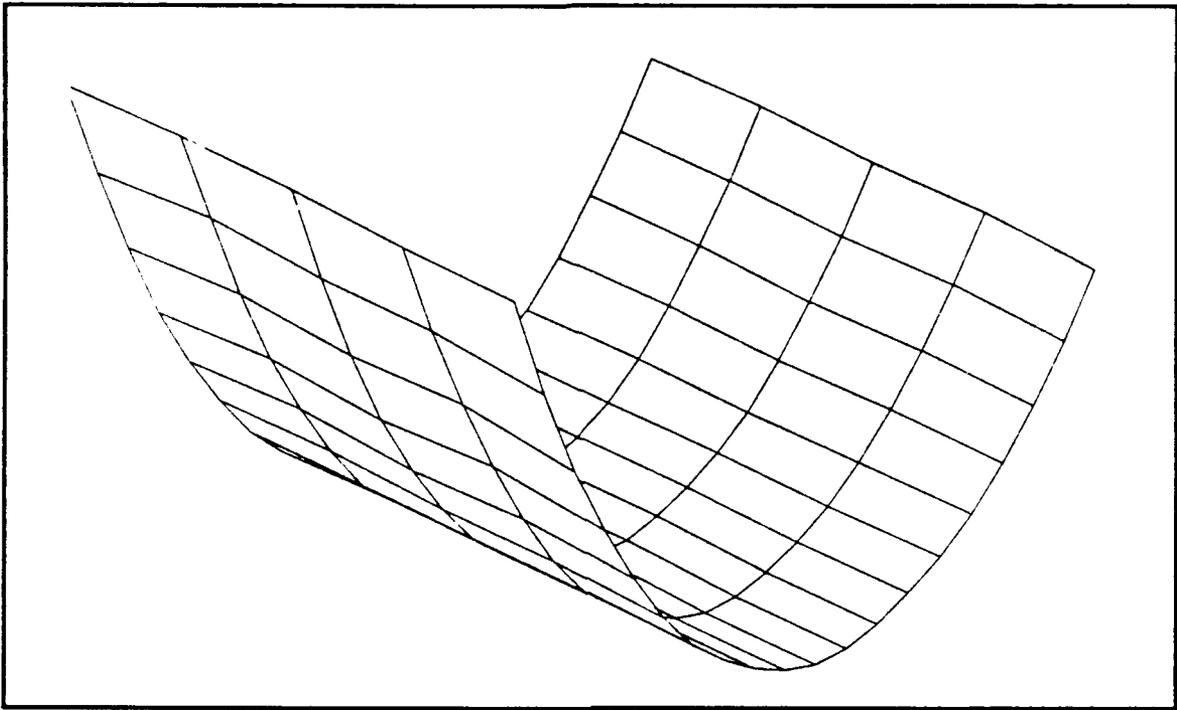


Fig. 5 Digitized Concave Surface

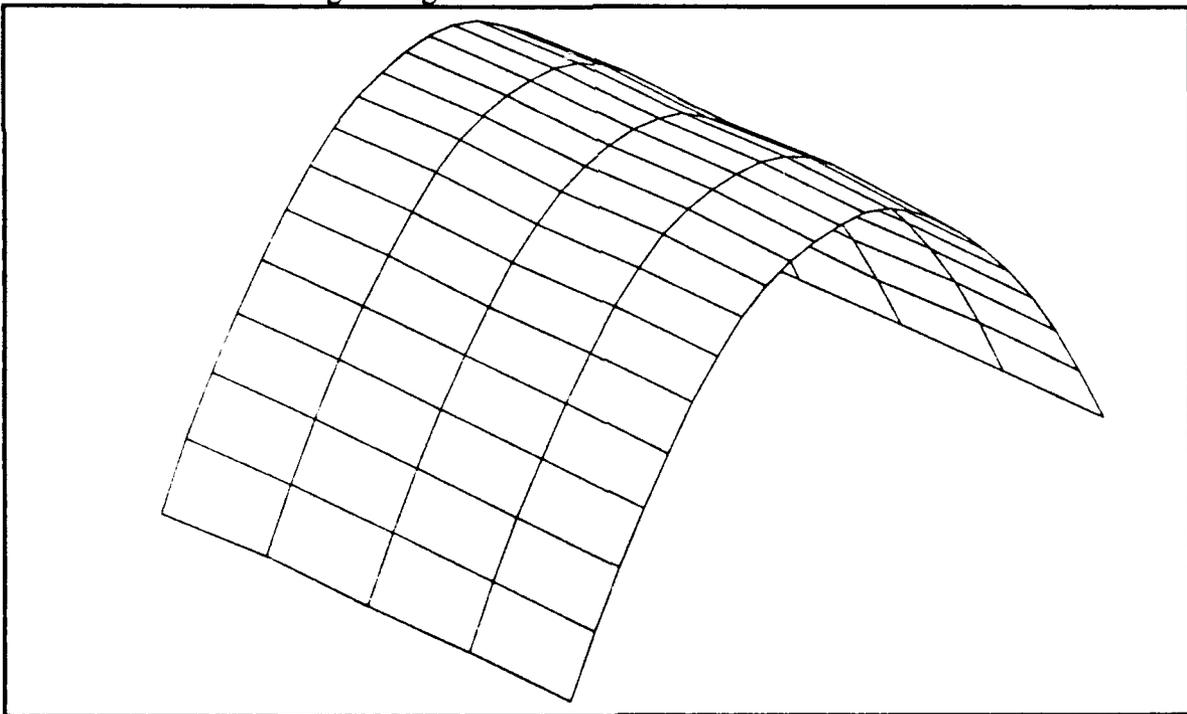


Fig. 6 Digitized Convex Surface

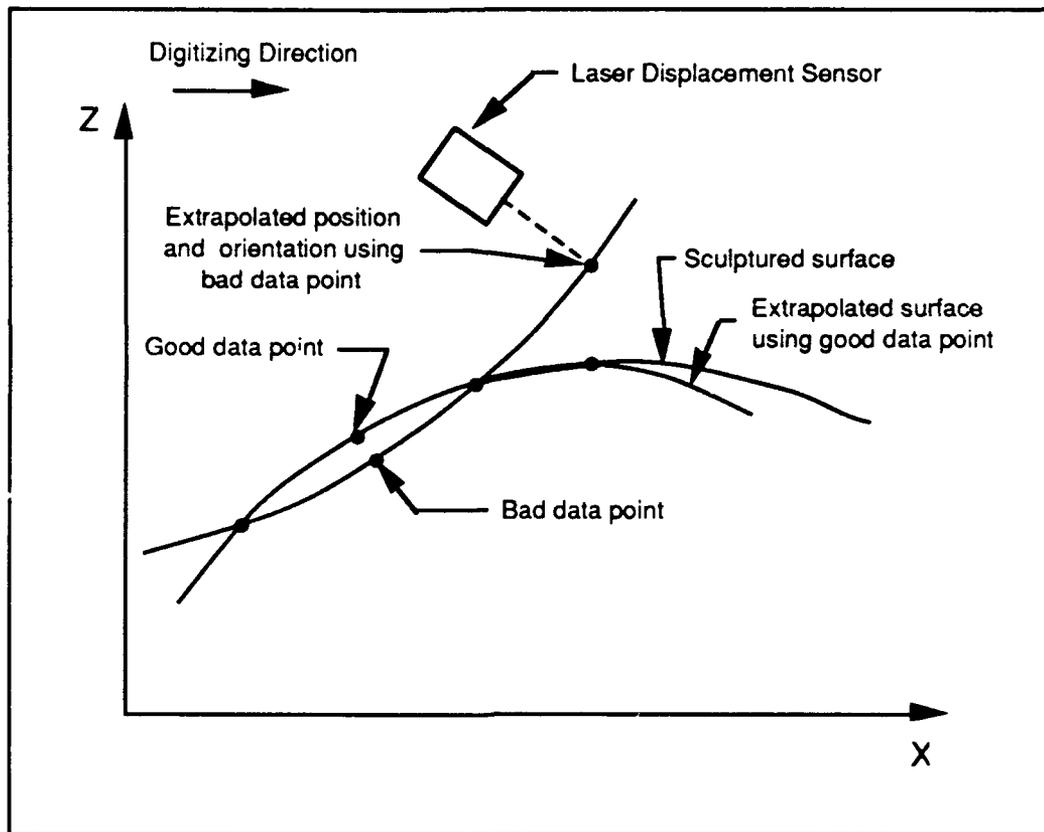


Fig. 7 2nd Order Polynomial Extrapolation with Bad Data Point

Note that changes in the middle point changes the predicted surface from concave to convex. For comparison, a 1<sup>st</sup> order polynomial is shown using the same data points.

#### D. Non-linear Reorientation Path

The robot's controller automatically determines the path the end-effector will take when moving from point to point. The path taken normally keeps the laser beams in a plane parallel to the xz plane. This way, the end-effector moves in a near linear direction. However, if the robot's wrist joints have reached their limit, the controller will rotate joints back 180° to acquire the desired configuration. During this re-orientation the end-effector moves in a circular path. For some surfaces (such as concave surfaces), this circular path can cause the LDS assembly to collide with the surface.

Using LDS's with a large stand-off distance helps to prevent the LDS assembly from colliding with the surface. Also, the radius that the end-effector travels in when re-orienting can be minimized by digitizing with the robot's first and second joint axes oriented perpendicular to the "world" x axis. This method works when digitizing small surfaces. However, this configuration

cannot be maintained when digitizing large surfaces. Another approach is to raise the end-effector to clear the surface before changing the end-effector's orientation. With the end-effector in the proper orientation, it can then be moved linearly to the next digitizing position. However, this clearance move would considerably slow down the digitizing rate. Finally, the path that the end-effector takes can be closely controlled by programming in joint axis coordinates rather than in Cartesian coordinates. This method requires the control module on the IBM PC to be more complicated and it also makes the control-module machine dependent for the robot that is being used.

### **E. LDS Reading Errors**

The LDS failed to provide correct displacement readings due to surface reflectivity and to geometry obstructions. The reflectivity of the surface effected the LDS's ability to function properly. If a surface is highly reflective and the LDS is oriented so that the laser beam is reflected away from the light sensor (CCD) in the LDS then an error reading will result.

Also, certain geometries can obscure the laser beam from returning to the CCD sensor, (such as a step, or a sphere). For this experiment, the LDS was kept perpendicular to the surface in only the xz plane. Therefore, in general, the LDS is not kept normal to the surface. This can cause the laser beam to be oriented in a bad position when measuring a sphere. In the laboratory, a basketball was digitized. Measurements close to the center of the ball were good. However, on the extreme y axis locations on the sphere, the laser beam was completely obstructed from returning to the CCD sensor on the LDS.

## **5. CONCLUSIONS**

We have described a new approach for digitizing unknown surfaces. Research results demonstrated that the robotic non-contact 3-D digitizing system and the multiple beam approach are effective for digitizing an unknown surfaces. This research illustrated that more general surfaces can be digitized using a six degrees-of-freedom robot to position and orient the LDS than can be digitized using a three axis coordinate measuring machines.

Another advantage of our approach is that the digitizing system was built on feasibly available equipment, such as, robot manipulators and a micro-computers. This demonstrated that a robust and precise digitizing system can be built at a low cost.

As discussed in Introduction section, keeping the laser beam perpendicular to the surface is a key factor that affects the precision and robustness of the digitizing process. By using the

approaches described in this report, namely multiple beam and extrapolation approaches, the highest possible LDS precision is obtained during the digitizing process. However, the precision of the measurement is also affected by the resolution and precision of the robotic system. Therefore, careful calibration of the robotic system is necessary. Robotic system calibration was studied previously, but not in the context of the robot being used as a measurement device [4, 5, 6]. The overall precision of robotic digitizing systems remains to be studied.

Typically, digitizing a surface produces more data points than required to describe the surface. In this case, the data points can be lumped together to create a mathematical model to describe the surface. The mathematical surface models provides a more flexible and efficient way of describing the surface. More work, however, needs to be done to automate the process of going from digitized data points to a mathematical surface model.

## REFERENCES

- [1] D. J. Porembiak, Private Communication, General Motor Corporation, Advanced Development Center, September, 1991.
- [2] K. G. Harding, "Sensors for the '90s," *Manufacturing Engineering*, April 1991, pp. 57-61.
- [3] K. Saito, et al., "Noncontact 3-D Digitizing and Machining System for Free-Form Surfaces," *Annals of the CIRP*, vol. 40, p. 483-486, 1991.
- [4] J. Chen and L. M. Chao, "Positioning error analysis for robot manipulators with all rotary joints," *IEEE J. of Robotics and Automation*, vol. RA-3, no. 6, p. 539-545, Dec. 1987.
- [5] J. H. Borm and C. H. Meng, "Determination of optimal measurement configurations for robot calibration based on observability measure," *Int. J. of Robotics Research*, vol. 10, no. 1, p. 52-63, Feb. 1991.
- [6] S. A. Hayati, "Robot arm geometric link parameter estimation," *Proc. 22nd IEEE CDC*, San Antonio, Texas, p. 1477-1483, Dec. 1983.

## APPENDIX B

# A New Robotic Hand/Eye Calibration Method by Active Viewing of a Checkerboard Pattern<sup>1</sup>

Chichyang Chen and Yuan F. Zheng  
Department of Electrical Engineering  
The Ohio State University  
2015 Neil Avenue  
Columbus, OH 43210

### Abstract

A simpler robotic hand/eye calibration method is developed in this study contrasted to the conventional methods which require a complicate implementation of camera calibration. First, the optical axis of the camera and the transformation between the target coordinate system and the world coordinate system are found by using active movement of the camera. Secondly, camera calibration using two-plane model is performed. The image processing techniques required for these two steps are developed, which are simpler and efficient. The implementation of the camera calibration using two-plane model and the image processing techniques indicate the efficiency and simplicity of the proposed hand/eye calibration method. The simulation and experimental results are reported.

---

<sup>1</sup>Acknowledgment: This work is supported by the Office of Naval Research under grant N00014-90-J-1516

## I. Introduction

In 3D vision, eye-on-the-hand configuration is very useful in many applications, such as, automated 3D robotics vision measurement, automated sensor placement planning, automated part acquisition or assembly, and stereo vision [1]. Essential in using this configuration is the hand/eye calibration, that is, the measurement of the relative 3D position and orientation between the camera and the robot gripper. This measurement has to be made so that the vision sensor can be moved to the desired locations to sense the environment or the object from different views. And, in reverse, the scene seen through the camera has to be interpreted in the robot's coordinate system since the scene seen through the camera is only relative to the camera.

The difficulty in hand/eye calibration is to resolve the nonlinearity caused by the rotations between the coordinate systems of the gripper and the camera. Puskorius and Feldkamp developed a method that can simultaneously calibrate the kinematics and the hand/eye relation of the robot [2]. However, their method needs to solve a large set of non-linear equations, which requires good initial values of the solution and is very time-consuming. Shiu and Ahmad proposed another approach which only dealt with the hand/eye calibration [3]. The calibration is formulated as solving the following equation:

$$AX = XB, \tag{1}$$

where  $A$  is the movement of the gripper,  $B$  is the resulting displacement of the camera, and  $X$  is the unknown transformation between the camera and the robot wrist. By treating cos and sin functions as independent variables, only linear equations need to be solved, but the number of variables are doubled. Recently, Tsai and Lenz have developed an approach for hand/eye calibration [1] similar to the approach proposed

by Shiu and Ahmad. However, by using Tsai and Lenz's method, the number of variables will not increase and only linear equations need to be solved. High accuracy and efficiency of their method have been claimed in their study.

All those methods mentioned above require obtaining the relative pose of the vision sensor to a fixed target block. The target block acts as an intermediate stage in the process of hand/eye calibration and can also provide redundant frames to improve the accuracy of the hand/eye calibration. In other words, these methods all require to do camera calibration with respect to the coordinate frame of the fixed target block. Both Tsai [1] and Shiu et. al. [3] deduced a conclusion that the camera needs to be moved to at least three different poses in order to obtain the desired transformation between the camera coordinate system and the robot gripper's coordinate system. Thus, the accurate and efficient performance of the the hand/eye calibration relies on the performance of the camera calibration.

To obtain the accuracy of camera calibration is a difficult task that usually requires professional expertise and expensive equipment which may hinder the popular use of vision sensors. The difficulty is mostly due to the nonlinearity of the lens distortion. Two kinds of camera model are commonly used for camera calibration, the pinhole model and the two-plane model. The methods using pinhole model require to derive the intrinsic camera parameters, such as the focal length, scale factors, lens distortion parameters; and the extrinsic parameters including the rotation angles and translation vector between the camera coordinate system and the world coordinate system. The transformation between the image frame of the camera and the world coordinate system is constructed from these parameters. A few of these methods can be seen in [4], [5], and [6]. The disadvantages of these methods using pinhole model are the complexity of implementation and the requirement of prior knowledge of some camera parameters. For example, in Tsai's method [6], the  $x$  scale factor (the center to center distances between adjacent sensor elements in  $x$  direction) of the camera is needed.

Although his method is reported to be very accurate, the accuracy depends on the accuracy of the scale factor. It is not always easy to obtain the parameter. And the parameter is not always reliable. Furthermore, Tsai's method is still complicate to be implemented, compared to the methods using two-plane model.

From user's point of view, two-plane model is probably the simplest way to implement camera calibration. It does not require to derive any camera parameters, i.e. knowledge of any camera parameters are not prerequisite. It only requires to set up the correspondence between the calibration points on the two parallel target blocks and their projected points in the image frame. The camera calibration methods using two-plane model can be seen in [7], [8], and [9]. Unfortunately, the two-plane method can not directly be applied to the methods for hand/eye calibration that are mentioned above, since explicit transformation between the coordinate systems of the camera and the target block can not be obtained from the two-plane model.

A method proposed in this paper dealing with robotic *hand/eye calibration* is adopting the two-plane camera model. Unlike the methods [1], [2], and [3], in which the camera is passively moved to each different pose before performing the camera calibration, we first try to find the optical axis of the camera in the coordinate space of the gripper and the transformation between the coordinate system of the target block and the world coordinate system. As shown in the sequel, the derivation of the optical axis and the target block can be easily accomplished by actively moving the hand (the gripper) in the  $x$  and  $z$  directions with the same orientation in the gripper and to two different poses. After the optical axis of the camera and the target block are found, camera calibration can be performed by using the two-plane model in the gripper's coordinate system. Thus, the geometric relation between the camera and the gripper is established and the scene seen through the camera can be interpreted in the gripper's coordinate system. There are several advantages in this proposed method. First of all, the implementation is very simple in the consideration

of deriving the optical axis of the camera and the two-plane camera calibration. Secondly, only a small portion of the image frame needs to be processed to find the optical axis of the camera. Image processing can thus be performed very fast. Finally, the camera calibration methods using the two-plane model can be applied to hand/eye calibration. Users do not need to implement the complicate camera calibration by using the pinhole camera model.

The method described in Section II is to find the optical axis of the camera in gripper's coordinate space and the transformation between the coordinate system of the target block and the world coordinate system. The image processing techniques used in our work are outlined in Section III. Simulation and experimental results are shown in Section IV. Section V concludes this paper.

## II. Derivation of the Optical Axis and the Target block

The base coordinate system of the robot is considered to be the robot world coordinate system. The target block is placed in a plane parallel to the  $x$ - $y$  plane of the world coordinate system. Shown in Fig. 1 is an illustration of the setup for the hand/eye calibration. The target block is designed to be a checkerboard colored with three distinct gray levels so that the intersection of the optical axis of the camera and the target block can be identified. The design of the target block is described in details in Section III.

The optical axis of the camera, line  $OA$  in Fig. 1, is expressed in parametric form in gripper's coordinate system as:

$$\begin{cases} x_g = x_{oa} + lt \\ y_g = y_{oa} + mt \\ z_g = z_{oa} + nt, \end{cases} \quad (2)$$

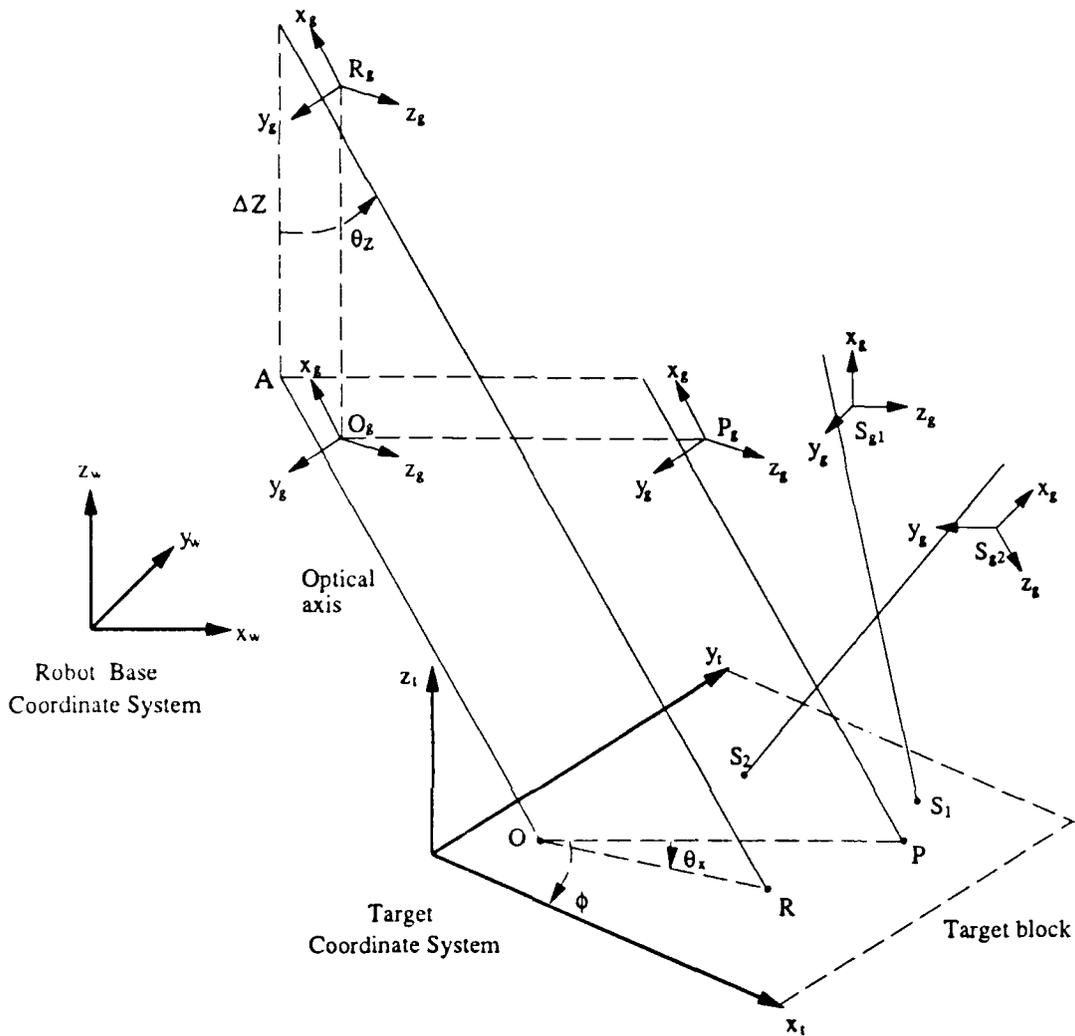


Figure 1: An illustration of the setup for hand/eye calibration.

where  $l, m, n$  are the direction cosines of the optical axis, and  $[x_{oa} \ y_{oa} \ z_{oa}]^T$  is a fixed point on the axis, which can be arbitrarily chosen along line  $OA$ . Since the target block is parallel to the  $x$ - $y$  plane of the robot world coordinate frame, there is only a rotation about  $z$  axis and a translation from the world coordinate system to the target coordinate system. We assume that the rotation angle is  $\phi$  and the translation is  $[T_x \ T_y \ T_z]^T$ .

To derive the rotation angle  $\phi$ , the gripper is first moved to a pose,  $O_g$ , so that its optical axis intersects the target block. For convenience, this intersection is simply referred to as “intersection point” at pose  $O_g$ . In Fig. 1, the intersection point is point  $O$ . Then, the gripper is moved in the  $x$  direction in robot world space with an arbitrary distance and without changing its orientation. We denote this pose as  $P_g$  and the intersection point at pose  $P_g$  as point  $P$ . Since the orientation of the gripper is not changed, line  $OP$  is parallel to the  $x$  axis of the robot world coordinate system. Thus the angle  $\phi$  of the rotation between the robot world coordinate system and the target coordinate system is the angle between line  $OP$  and the  $x$  axis of the target frame.

To derive  $l, m, n$ , the gripper is moved to a pose  $R_g$  that is along the  $z$  direction from pose  $O_g$  and has the same orientation as the pose  $O_g$ , as shown in Fig. 1. The intersection point at pose  $R_g$  is denoted as  $R$ . The direction cosines  $l_w, m_w, n_w$ , of the optical axis in the robot world coordinate system can then be determined from the following geometric information: the angle  $\theta_x$  between line  $OR$  and line  $OP$ , the distance  $\Delta z$  between pose  $O_g$  and pose  $R_g$ , and the distance  $\|OR\|$  between point  $O$  and point  $R$ . That is,

$$\begin{aligned} l_w &= \sin(\theta_z) \cos(\theta_x), \\ m_w &= \sin(\theta_z) \sin(\theta_x), \\ n_w &= -\cos(\theta_z), \end{aligned} \tag{3}$$

where  $\theta_z = \tan^{-1} \frac{(\Delta z)}{\|OR\|}$ . The orientation  $n_0, s_0, a_0$  of the gripper can be read from the robot controller. Thus, the direction cosines  $l, m, n$  of the optical axis in the gripper's coordinate system are then,

$$\begin{cases} l &= [l_w \ m_w \ n_w]^T \cdot n_0 \\ m &= [l_w \ m_w \ n_w]^T \cdot s_0 \\ n &= [l_w \ m_w \ n_w]^T \cdot a_0. \end{cases}$$

$x_{oa}$ ,  $y_{oa}$ , and  $z_{oa}$  remain to be found. Since the point  $[x_{oa} \ y_{oa} \ z_{oa}]^T$  can be chosen arbitrarily along the optical axis, one variable can be reduced by setting one of the three variables to zero. For example, we can set  $x_{oa}$  to zero. Then the gripper is moved to two different poses  $S_{g_1}$  and  $S_{g_2}$ . The intersection points at these two poses are denoted as  $S_1$  and  $S_2$  with coordinates  $[X_1 \ Y_1 \ 0]_t^T$  and  $[X_2 \ Y_2 \ 0]_t^T$ , respectively, in the target coordinate system. Note that it is assumed the target block is at the plane  $z_t = 0$  in the target frame.  $y_{oa}$  and  $z_{oa}$  can then be derived from the constraint that  $O$ ,  $S_1$ , and  $S_2$  are on the target block, with their coordinates  $[X_0 \ Y_0 \ 0]_t^T$ ,  $[X_1 \ Y_1 \ 0]_t^T$ , and  $[X_2 \ Y_2 \ 0]_t^T$  known. The coordinates  $[x_0 \ y_0 \ z_0]^T$ ,  $[x_1 \ y_1 \ z_1]^T$ , and  $[x_2 \ y_2 \ z_2]^T$  of the points  $O$ ,  $S_1$ ,  $S_2$  in the robot world coordinate system can be derived from

$$\begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = A_i \begin{bmatrix} lt_i \\ y_{oa} + mt_i \\ z_{oa} + nt_i \\ 1 \end{bmatrix}, \quad (4)$$

where  $i = 0, 1, 2$ , and  $A_0$ ,  $A_1$ , and  $A_2$  are the transformation matrices from the base of the robot to the pose of the gripper at  $O_g$ ,  $S_{g_1}$ , and  $S_{g_2}$ , respectively.  $[lt_i \ y_{oa} + mt_i \ z_{oa} + nt_i]^T$ ,  $i = 0, 1, 2$  are the coordinates of  $O$ ,  $S_1$ , and  $S_2$  in the gripper's coordinate system.

Since the rotation angle  $\phi$  of the target coordinate system is already known, we should have

$$\begin{aligned} x_i - x_0 &= (X_i - X_0) \cos(\phi) - (Y_i - Y_0) \sin(\phi), \\ y_i - y_0 &= (X_i - X_0) \sin(\phi) + (Y_i - Y_0) \cos(\phi), \\ z_i - z_0 &= 0. \end{aligned} \quad (5)$$

For each pose  $S_{g_i}$  of the gripper, three conditions are available to solve  $y_{oa}$  and  $z_{oa}$ . However, one variable  $t_i$  is also added. Totally, for two poses  $S_{g_1}$  and  $S_{g_2}$ , there are six conditions to solve five variables  $y_{oa}$ ,  $z_{oa}$ ,  $t_0$ ,  $t_1$ , and  $t_2$ . The solution

$X = [y_{0a} \ z_{0a} \ t_0 \ t_1 \ t_2]^T$  can be obtained by using least square error method. That is,

$$X = (B^T B)^{-1} B^T Y,$$

where matrix  $B$  and vector  $Y$  can be derived easily from equations (4) and (5) to be

$$B = \begin{bmatrix} a_{12}^1 - a_{12}^0 & a_{13}^1 - a_{13}^0 & -(a_{11}^0 l + a_{12}^0 m + a_{13}^0 n) & a_{11}^1 l + a_{12}^1 m + a_{13}^1 n & 0 \\ a_{22}^1 - a_{22}^0 & a_{23}^1 - a_{23}^0 & -(a_{21}^0 l + a_{22}^0 m + a_{23}^0 n) & a_{21}^1 l + a_{22}^1 m + a_{23}^1 n & 0 \\ a_{32}^1 - a_{32}^0 & a_{33}^1 - a_{33}^0 & -(a_{31}^0 l + a_{32}^0 m + a_{33}^0 n) & a_{31}^1 l + a_{32}^1 m + a_{33}^1 n & 0 \\ a_{12}^2 - a_{12}^0 & a_{13}^2 - a_{13}^0 & -(a_{11}^0 l + a_{12}^0 m + a_{13}^0 n) & 0 & a_{11}^2 l + a_{12}^2 m + a_{13}^2 n \\ a_{22}^2 - a_{22}^0 & a_{23}^2 - a_{23}^0 & -(a_{21}^0 l + a_{22}^0 m + a_{23}^0 n) & 0 & a_{21}^2 l + a_{22}^2 m + a_{23}^2 n \\ a_{32}^2 - a_{32}^0 & a_{33}^2 - a_{33}^0 & -(a_{31}^0 l + a_{32}^0 m + a_{33}^0 n) & 0 & a_{31}^2 l + a_{32}^2 m + a_{33}^2 n \end{bmatrix} \quad (6)$$

and

$$Y = \begin{bmatrix} a_{14}^0 - a_{14}^1 + (X_1 - X_0) \cos(\phi) - (Y_1 - Y_0) \sin(\phi) \\ a_{24}^0 - a_{24}^1 + (X_1 - X_0) \sin(\phi) + (Y_1 - Y_0) \cos(\phi) \\ a_{34}^0 - a_{34}^1 \\ a_{14}^0 - a_{14}^2 + (X_2 - X_0) \cos(\phi) - (Y_2 - Y_0) \sin(\phi) \\ a_{24}^0 - a_{24}^2 + (X_2 - X_0) \sin(\phi) + (Y_2 - Y_0) \cos(\phi) \\ a_{34}^0 - a_{34}^2 \end{bmatrix}, \quad (7)$$

where  $a_{ij}^0$ ,  $a_{ij}^1$ , and  $a_{ij}^2$  are the elements of the matrices  $A_0$ ,  $A_1$ , and  $A_2$ , respectively.

After obtaining the coordinates of the intersection points  $O$ ,  $S_1$ , and  $S_2$  in both the robot world and target coordinate systems, the translation of the target coordinate system can be easily derived from the following equation:

$$\begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 & T_x \\ \sin(\phi) & \cos(\phi) & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ 0 \\ 1 \end{bmatrix},$$

$i = 0, 1, 2$ . That is,

$$\begin{aligned}
T_x &= x_i - \cos(\phi)X_i + \sin(\phi)Y_i, \\
T_y &= y_i - \sin(\phi)X_i - \cos(\phi)Y_i, \\
T_z &= z_i.
\end{aligned} \tag{8}$$

More accurate values of  $T_x$ ,  $T_y$ , and  $T_z$  can be obtained by using least square error method from the three sets of data  $[x_i, y_i, z_i]$  and  $[X_i, Y_i, 0]$ , for  $i = 0, 1, 2$ .

### III. Image Processing Techniques

There are two image processing tasks needed to be accomplished for the proposed robotic hand/eye calibration method. The first task is to obtain the position of the intersection point on the target block. The second task is to extract the points in the image frame projected from the set of  $n$  calibration points on the target block, which are used in the process of two-plane camera calibration.

To accomplish the first task, the scene seen through the camera need to be recognized so that the position of the intersection point can be identified. In our implementation, the target block is designed to be a  $6 \times 6$  evenly spaced checkerboard. Each square on the checkerboard is colored with one of three different gray levels. The squares are colored in a way to make the gray pattern of the neighboring squares (up, down, left, and right neighbors) of each square unique. Three different gray levels are chosen as dark black, pure white, and gray so that their gray levels in the image are distinct. Fig. 2 shows such a target block. Greater number of gray levels should be used if the size of the target block is larger.

The first task includes two works. The first is to recognize the square in which the intersection point is. For convenience, this square is referred to as "center square" hereafter, since the square is always at the center of the image frame. The second is

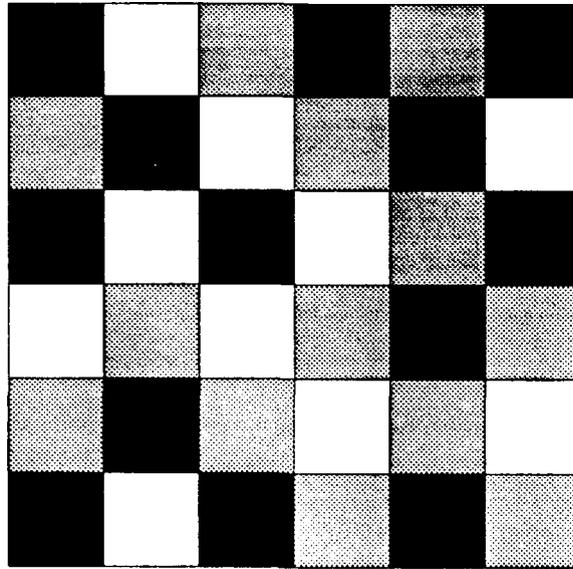


Figure 2: The target block.

to compute the position of the intersection point within that square. The procedure to accomplish the first task includes the following steps:

*1. Set up a processing window*

Since we are only interested in detecting the position of the intersection point, only the neighboring area of the image center needs to be processed. Thus the first step is to set up a window centered at the image center in the image frame. The size of this window must be large enough to cover the image of  $2 \times 2$  squares on the target block. The size of this window can be determined empirically.

*2. Threshold the image in the window*

Threshold the window into a three-level image. The threshold values can either be computed automatically by moment-preserving thresholding [10] or be determined empirically.

*3. Extract the boundaries of the center square*

To simplify the problem of extracting the boundaries, the camera is moved to a

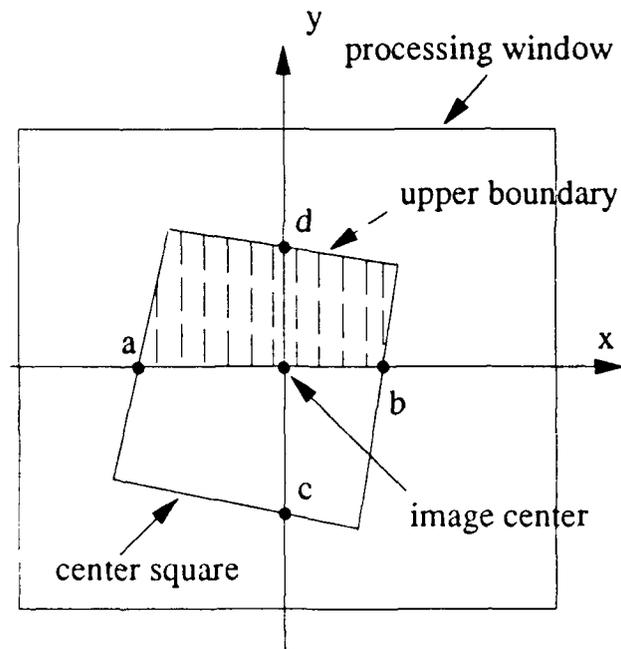


Figure 3: Finding the upper boundary of the center square.

pose so that the tilt of the image projected from the center square is small enough and each of the four corners of the center square in the image is in each of the four quadratures, as shown in Fig. 3. First, along  $x$  and  $y$  axes of the image frame, detect points  $a$ ,  $b$ ,  $c$ , and  $d$  shown in Fig. 3 by using one-dimensional difference operations. These points are the boundary points of the square on the  $x$  and  $y$  axes of the image frame. Secondly, perform one-dimensional difference operations along the lines that are perpendicular to the line  $ab$  and pass through every pixel on the line  $ab$  in the positive  $y$  direction. Then find out the edge point with maximum  $y$  value. In the case shown in Fig. 3, the edge points on the right side of the point with maximum  $y$  value are the upper boundary points. The edge points on the lower, right, and left boundaries of the center square can be found in a similar manner.

4. Find out the corners of the center square

The equations of the four boundaries of the center square can be derived from

their edge points by using least square error method. The corners of the square can then be computed to be the intersections of these four boundaries.

*5. Determine the position of the intersection point within the center square*

Since the image of a square is distorted by the perspective projection of the camera, we have to compute two vanishing points. These two points are the intersections of the two pairs of lines projected from the parallel boundaries of the square. Then compute the intersection  $e$  of the line passing through the image center and the vanishing point  $g$  with the lower boundary of the square and the intersection  $f$  of the line passing through the image center and the vanishing point  $h$  with the left boundary of the square, as shown in Fig. 4. The position of the image center within the image of the center square can then be determined by these two intersections,  $e$  and  $f$ . Since the image center is actually the projection of the intersection point into the image frame, we have thus determined the position of the intersection point within the center square.

*6. Identify the position of the center square*

Examine the gray-level of the center square and the squares adjacent to the four boundaries of the center square. The position of the center square within the target block can then be determined by the coloring pattern of these squares.

The implementation of the second task is relatively easier, compared to the first task. The calibration points are designed to be the center points of all the squares on the target block. As the camera is moved to a pose so that the optical axis of the camera is perpendicular to the target block and the image of the camera covers the whole target block, the camera acquires the image. Then, the following steps are performed:

1. Threshold the image as done in step 2 for task one.
2. Extract the boundaries of the center square as performed in step 3 for task one.

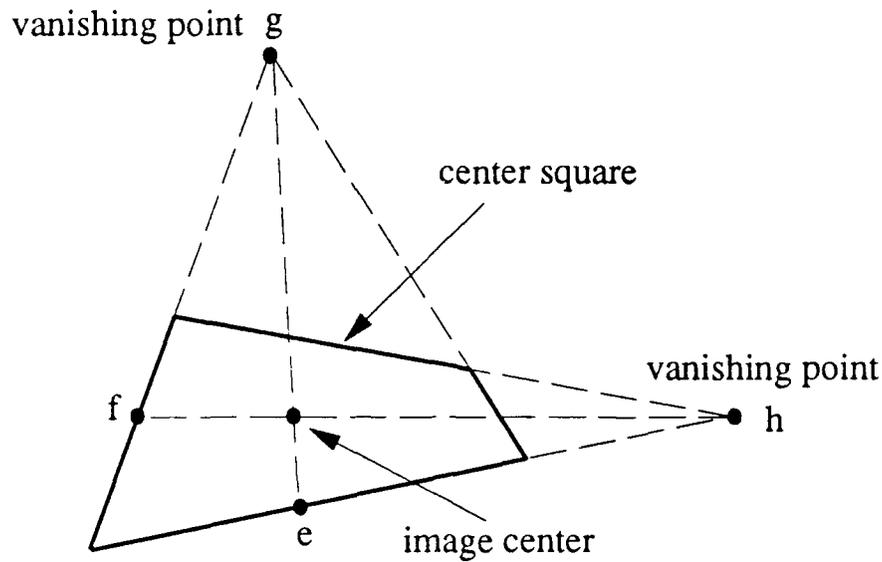


Figure 4: Determination of the position of the image center within the center square.

After this step, the directions of the four boundaries are known.

3. Because the target block is parallel to the image plane, the direction of each boundary of all the squares other than the center square in the image is the same as the direction of the corresponding boundary of the center square. The boundaries of all the squares in the image can then be easily detected using one-dimensional difference operations.
4. Compute the center points of all the squares in the image from the detected boundaries. These points can then be used in the process of camera calibration using two-plane camera model.

## IV. Simulation and Experimental Results

Since the experimental results of the camera calibration using two-plane model have been reported to be very accurate [9], only the effectiveness of our methods in finding the position and orientation of the camera's optical axis and the transformation of the target block is tested. The description and the results of the simulations and experiments are given in the following.

### A. Simulations

Two kinds of simulations were made. One of them is based on the assumption that the target block is exactly parallel to the  $x$ - $y$  plane of the robot world coordinate system. The other kind assumes that there are small rotations in the  $x$  and  $y$  directions between the two coordinate systems. The setup for the simulations is designed to be the same as the real setup for the experiments.

We first assumed that the translation from the world coordinate system to the coordinate system of the target block is  $[37.82cm \ 38.57cm \ -66.29cm]^T$  and the rotation angle with  $z$  axis is 1.324 degree between these two systems. The optical axis of the camera, in gripper's coordinate system, is assumed to be

$$\begin{cases} x_g = 0.9987t \\ y_g = 0.24cm + 0.0507t \\ z_g = 6.84cm + 0.0099t, \end{cases}$$

These data about the setup for the simulations are obtained from the experiments described in the Section IV-B.

Uniformly distributed noises are added to the position and orientation of the robot gripper that are read from the robot controller to account for the inaccuracy of the robot arm. Uniformly distributed noises are also added to the position of the

Table 1: Simulation results with the target block parallel to the  $x$ - $y$  plane of the robot world coordinate system (the unit for  $y_0$ ,  $z_0$ ,  $T_x$ ,  $T_y$  and  $T_z$  is  $mm$  and the unit for  $l$ ,  $m$ ,  $n$  is  $10^{-3}$ ).

$M$	$N$	<i>Max. error</i>								
		$l$	$m$	$n$	$y_0$	$z_0$	$\phi$	$T_x$	$T_y$	$T_z$
1	2	0.792	14.762	16.854	5.016	6.419	0.446°	5.371	3.052	2.333
2	3	0.353	7.157	3.645	1.399	2.434	0.249°	2.354	0.370	0.608
3	4	0.284	5.062	2.557	1.304	2.576	0.119°	2.386	0.427	0.580
4	5	0.232	4.661	1.541	1.354	1.626	0.097°	1.530	0.489	0.130
5	6	0.173	3.437	1.745	0.802	0.819	0.081°	0.741	0.168	0.113

Table 2: Simulation results with the target block parallel to the  $x$ - $y$  plane of the robot world coordinate system (the unit for  $y_0$ ,  $z_0$ ,  $T_x$ ,  $T_y$  and  $T_z$  is  $mm$  and the unit for  $l$ ,  $m$ ,  $n$  is  $10^{-3}$ ).

$M$	$N$	<i>Avg. error</i>								
		$l$	$m$	$n$	$y_0$	$z_0$	$\phi$	$T_x$	$T_y$	$T_z$
1	2	0.252	4.928	4.405	1.236	1.605	0.123°	1.547	0.790	0.777
2	3	0.120	2.275	1.023	0.456	0.727	0.068°	0.691	0.085	0.185
3	4	0.081	1.551	0.601	0.347	0.650	0.038°	0.644	0.138	0.143
4	5	0.063	1.220	0.440	0.368	0.410	0.029°	0.408	0.144	0.036
5	6	0.050	0.974	0.431	0.228	0.217	0.022°	0.195	0.047	0.027

intersection points ( $x$  and  $y$  coordinates in the target coordinate system) to account for the errors caused by the inaccuracy of image processing. The maximum noise of the gripper's position is assumed to be  $0.1mm$  and the maximum noise of the gripper's orientation (the  $O$ ,  $A$ ,  $T$  angles) is assumed to be  $0.05$  degrees. The maximum error in the position of the intersection point is assumed to be  $5\mu m$ .

Based upon these assumptions, we simulated the process of the hand/eye calibration two hundred times. The results of these simulations are shown in Table 1 and Table 2.

In Table 1 and Table 2,  $M$  is the number of times the movement of the camera in each direction of the world  $x$  and  $z$  axes,  $N$  is the number of poses that the camera was moved to derive the  $y_0$  and  $z_0$  in equation (2). The movement of the gripper in each direction of the  $x$  and  $z$  axes is designed in such a manner that the distance from the pose  $P_g$  and  $R_g$  to the pose  $O_g$  is increased incrementally from  $i = 1$  to  $i = 5$ .

Then simulations were made by imposing one more assumption that the target block was rotated by  $0.5$  degree about the  $x$  and  $y$  axes of the robot world coordinate system. The results are listed in Table 3 and Table 4.

### *B. Experiments*

Fig. 5 is the picture of the setup for the experiments. A TM-540 CCD camera was mounted on the end effector of a PUMA 560 robot arm. The resolution of the frame grabber is  $480 \times 512$ . For simplicity, the target block was colored only by black and white colors. The identification of the center square is done by human's eyes. The size of each square on the calibration plane is  $2.61cm \times 2.61cm$  with an accuracy of about  $0.2mm$  only.

Since it is impossible to have the exact equations of the optical axis and the target block, in order to verify the effectiveness of our method, a verification method similar to the method used in [1] was adopted and is described below. First, in the calibration

Table 3: Simulation results with the target block rotated  $0.5^\circ$  with the world  $x$  and  $y$  axes (the unit for  $y_0, z_0, T_x, T_y$  and  $T_z$  is  $mm$  and the unit for  $l, m, n$  is  $10^{-3}$ ).

$M$	$N$	<i>Max. error</i>								
		$y_0$	$z_0$	$l$	$m$	$n$	$\phi$	$T_x$	$T_y$	$T_z$
1	2	0.669	15.206	12.483	5.455	7.183	$0.601^\circ$	6.893	2.961	2.712
2	3	0.333	7.244	4.231	2.823	3.223	$0.431^\circ$	3.304	1.912	0.870
3	4	0.278	5.953	4.099	3.018	3.239	$0.388^\circ$	3.383	2.157	0.787
4	5	0.248	5.229	3.469	3.067	1.679	$0.355^\circ$	1.498	2.225	0.842
5	6	0.228	4.850	3.327	1.775	1.583	$0.343^\circ$	1.256	1.128	0.868

Table 4: Simulation results with the target block rotated  $0.5^\circ$  with the world  $x$  and  $y$  axes (the unit for  $y_0, z_0, T_x, T_y$  and  $T_z$  is  $mm$  and the unit for  $l, m, n$  is  $10^{-3}$ ).

$M$	$N$	<i>Avg. error</i>								
		$y_0$	$z_0$	$l$	$m$	$n$	$\phi$	$T_x$	$T_y$	$T_z$
1	2	0.183	3.815	3.242	1.104	1.909	$0.269^\circ$	1.964	0.850	0.687
2	3	0.119	2.488	2.069	1.035	0.914	$0.266^\circ$	1.005	1.035	0.276
3	4	0.091	1.961	2.060	1.688	0.663	$0.258^\circ$	0.716	1.620	0.449
4	5	0.075	1.682	2.073	1.848	0.477	$0.263^\circ$	0.455	1.645	0.701
5	6	0.076	1.723	2.155	1.190	0.823	$0.265^\circ$	0.668	0.971	0.778

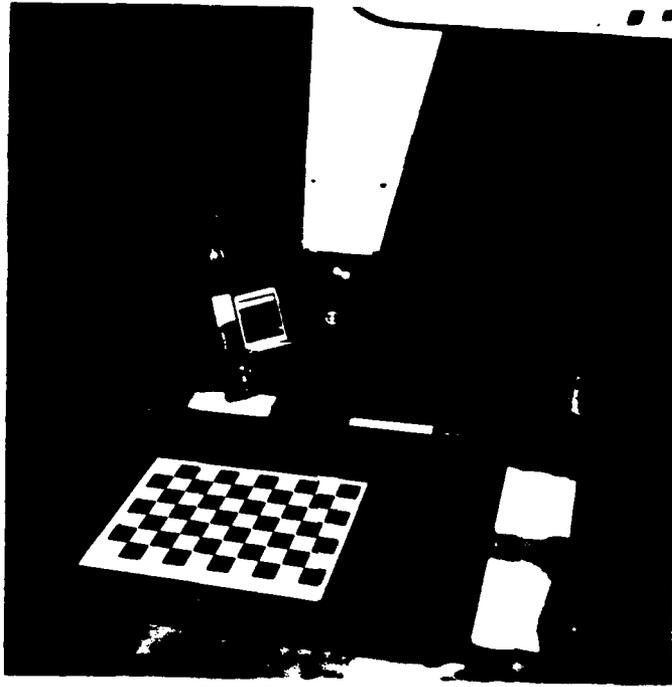


Figure 5: The setup for the experiments.

Table 5: Experimental results.

Avg. $d_{cp}$	$M$	$N$	$V$	Max. error	Avg. error
18.5cm	5	6	5	0.3453mm	0.2113mm
29.0cm	5	6	5	0.9272mm	0.5500mm

phase, the camera was moved  $M$  times in each direction of the  $x$  and  $z$  axes of the robot world coordinate system and  $N$  times to different poses. These movements were used to derive the equations of the optical axis and the target block. Then, in the verification phase, the camera was moved to  $V$  different poses and the positions of the intersection points at these  $V$  poses were recorded. The positions of these intersection points can also be computed by utilizing the equations of the optical axis and the target block obtained in the calibration phase. The distances between the recorded and the computed intersection points are regarded as the error of the calibration.

In order to test the effect of the distance between the camera and the target block on the accuracy of the calibration, the results of the experiments are categorized into two groups according to the distance between the camera and the target block at pose  $O_g$ . This distance is denoted as  $d_{cp}$ . The results are listed in Table 5.

### *C. Discussions*

From the results of the simulations and experiments, three factors that can influence the accuracy of the proposed hand/eye calibration method have been identified. They are outlined as follows:

1. The parallelism between the target block and the  $x$ - $y$  plane of the robot world coordinate system.
2. The distance of the movement of the intersection points in the  $x$  and  $z$  directions in the process of deriving the direction cosines of the optical axis of the camera.
3. The distance between the camera and the target block.

Comparing the simulation results in Table 1 and Table 2 to Table 3 and 4, we found that the parallelism between the target block and the  $x$ - $y$  plane of the robot world coordinate system has a dominating effect on the accuracy of the hand/eye calibration. Fortunately, this parallelism can be easily achieved by moving the robot arm in each direction of the  $x$  and  $y$  axes with a probe attached to the gripper and touching the surface of the calibration target block. The placement of the target block in the setup for our experiments is made in such a way.

The gripper will deviate from the desired orientation during its movement. Accordingly, error occurs in the position of the intersection points. The longer distance between the camera and the target block, the greater the error. This error

can be eliminated by decreasing the distance between the camera and the target block.

The error in the intersection points has a significant effect on the accuracy in deriving the direction cosines of the optical axis. The shorter the distance in the movement of the intersection point, the larger the error in deriving the rotation angle  $\phi$  of the target block and the  $\theta_x$  and  $\theta_z$  angles in equation (3). To have larger distance of the intersection points moving in the  $x$  direction, the amount of the movement of the gripper should be large. To have larger distance of the intersection point as the gripper is moved in the  $z$  direction, the optical axis should be oriented in a way that the angle ( $\theta_z$  in Fig. 1) between the optical axis and the world  $z$  axis is large.

We conclude that, to obtain better accuracy in the proposed hand/eye calibration method, the amount of the movement of the gripper in the  $x$  direction should be large, the angle  $\theta_z$  between the optical axis and the world  $z$  axis should be large (at pose  $O_j$ ), and the distance between the camera and the target block should be small.

## V. Conclusion

The robotic hand/eye calibration method proposed in this study is important in that it simplifies the process of robotic hand/eye calibration. Two-plane camera model provides a simpler implementation of camera calibration for this method. Compared to other methods [1, 2, 3], which require at least three times of camera calibration, this method needs only one performance of camera calibration to implement the hand/eye calibration. The algorithm developed in this method is simpler and easily to be understood. In addition, the image processing techniques are easier to be implemented and more efficient.

Autonomy has been paid attention in the literature [11] for evaluating the camera

calibration. The proposed hand/eye calibration method can allow the process of the hand/eye calibration be implemented automatically.

Finally, the simulation results tell that the proposed method should be very accurate in hand/eye calibration. The effectiveness of the proposed method has also been tested by experiments, although the experimental results are not very accurate due to the inaccuracy in making the target block.

## VI. References

- [1] Roger Y. Tsai and Reimar K. Lenz, "Real time versatile robotics hand/eye calibration using 3D machine vision," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1, pp. 554-561, April 1988.
- [2] G. Puskorius and L. Feldkamp, "Global calibration of a robot/vision system," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1, pp. 190-195, 1987.
- [3] Shiu, Y and Ahmad, S., "Finding the mounting position of a sensor by solving a homogeneous transformation equation of the form  $AX = XB$ ," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, pp. 1666-1671, 1987.
- [4] I. Sobel, "On calibrating computer controlled cameras for perceiving 3-D scenes," *Artificial Intelligence*, vol. 5, pp. 185-198, 1974.
- [5] Y. Yakimovsky and R. Cunningham, "A system for extracting three-dimensional measurements from a stereo pair of TV camera," *Computer Graphics and Image Processing*, vol. 7, pp. 195-210, 1978.

- [6] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the shelf TV cameras and lenses," *IEEE Journal on Robotics and Automation*, vol. RA-3, pp. 323-344, August 1987.
- [7] H. A. Martins, J. R. Birk, and R. B. Kelly, "Camera models based on data from two calibration planes," *Computer Graphics and Image Processing*, vol. 17, pp. 173-180, 1981.
- [8] A. Isaguirre, P. Pu, and J. Summers, "A new development in camera calibration: Calibrating a pair of mobile cameras," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1, pp. 74-79, 1985.
- [9] Keith D. Gremban, Charles E. Thorpe, and Takeo Kanade, "Geometric camera calibration using systems of linear equations," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1, pp. 562-567, April 1988.
- [10] W. H. Tsai, "Moment-preserving thresholding: A new approach," *Computer Vision, Graphics, and Image Processing*, vol. 29, pp. 377-393, 1985.
- [11] R. Tsai, *The Robotics Review 1*. Cambridge, MA: The MIT Press, 1989.

## APPENDIX C

### VIRTUAL SPACE OF DEFORMABLE OBJECTS HANDLED BY ROBOT END-EFFECTORS

Ming Z. Chen and Yuan F. Zheng  
Department of Electrical Engineering  
The Ohio State University  
Columbus, OH 43210

#### ABSTRACT

When a deformable object is moved by robot manipulator, the shape of the object varies. In order to avoid collision of the object with obstacles in the environment, the path of the object must be planned in accordance with the deformed shape of the object. To overcome uncertainty in the deformed shape, a virtual space approach is proposed in this paper. The virtual space is a hypothetical space which covers all the surfaces and edges of a deformable object. By using the virtual space, the deformable object can be treated as a rigid body. This will greatly simplify the path planning problem. To determine the virtual space of deformable objects, static deformations of objects handled by robot end-effectors are studied. The study investigates two cases: (1) deformations of the object as the end-effector pitches, and (2) deformations of the object as the end-effector rolls. Since no closed-form solution can be found for deformation-governing equations. A perturbation technique is used to obtain approximate solutions. Based on the deformation behavior of the object, the virtual space is determined using the minimal-volume and least-computation criteria.

---

ACKNOWLEDGEMENT: This work is supported by ONR under grant N00014-90-J-1516 and by NSF under a Presidential Young Investigator Award to Yuan F. Zheng.

## 1. INTRODUCTION

In the area of material handling, studies have been focused on rigid objects. In reality, non-rigid objects, such as composite materials, are frequently used in the manufacturing industry. In reality, these materials are customarily handled by human hands.

Recently, behavior of deformed objects becomes of considerable interest because of its relevance to problems encountered in the fields of spaceflight (spacecraft antennae, etc.) and robotics. In papers [1-3], equations of motion are formulated for a deformed body that executes motions relative to a rigid-body reference frame. In [4], Villarreal and Asada introduced the concept of Buffer Zones as a geometric representation of the effects of part compliance. With the use of the concept, the task of assembling a flexible 2-dimensional box is successfully planned and executed. In reference [5], Zheng, Pei and Chen proposed strategies for automatic assembly of deformable objects. The proposed strategies successfully completed the insertion of a flexible beam into a rigid hole. More recently, the authors of this paper proposed to coordinate two grippers to handle elastic objects [6].

When an object is moved from one place to another, one needs to plan the path of the object to avoid collision with obstacles in a crowded environment. Since the shape of a deformable object is uncertain, it is difficult to plan such a path. One convenient method is to treat the object as a rigid body by means of virtual space. The virtual space is a hypothetical space which covers all the surfaces and edges of an elastic object (Fig.1) regardless of the degree of deformation. As a result, the deformable object can be treated as a rigid body with the virtual space being the shape of the object. The complicated path planning problem thus becomes simple.

Selecting an adequate virtual space for a deformable object is not a simple issue. If a selected virtual space is too large, it is unable to make efficient use of the workspace. If too small, the object may collide with obstacles. The exact shape of the virtual space must be determined from the deformation behavior of the object, which may be both static and dynamic (vibration). Dynamic deformation is not considered as a main factor in this paper. This is because vibration is ignorable if the object is properly handled. In fact, vibration-free handling of deformable objects have been studied by many works (see [7] and its references).

As for static deformations, a large number of deflection analysis have been reported over the years. A summary of the studies may be found in [8] and [9]. The results as summarized in [8] and [9] cannot be directly applied in our case in which the objects are grasped by a robot end-effector which may alter its orientation. This is in contrast with the "stationary" boundary conditions as assumed by the works in [8] and [9]. As a result, one key point of our study is to determine static deformation of an object when the end-effector alters its orientation.

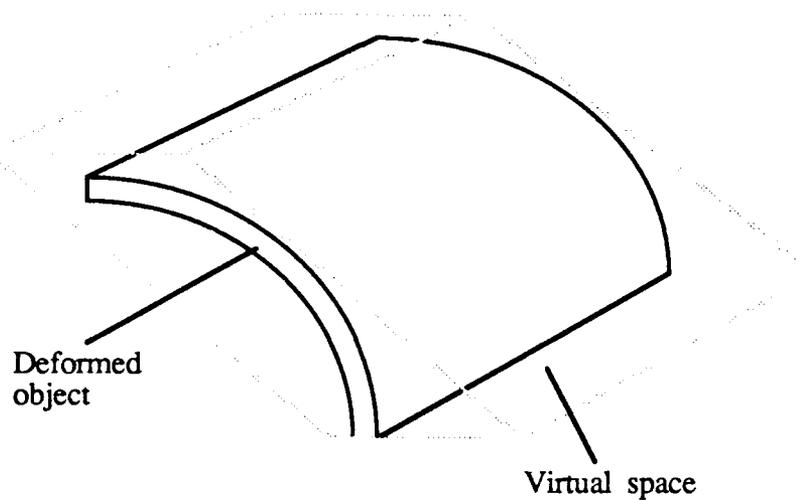


Fig. 1 The concept of the virtual space

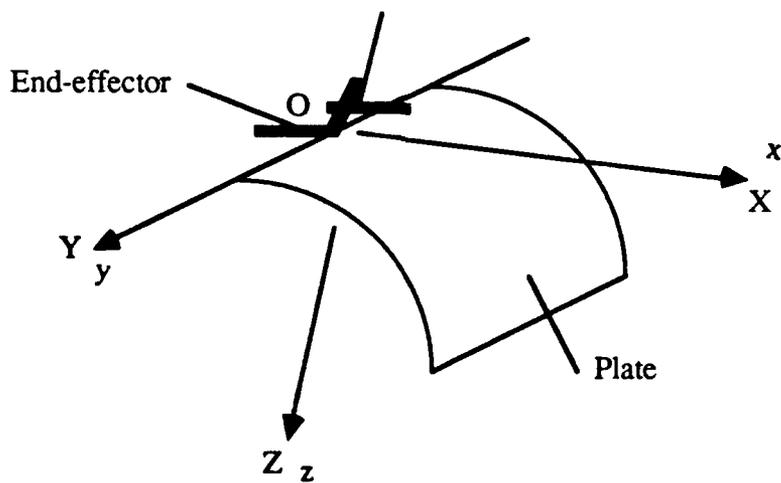


Fig. 2 A plate is grasped by an end-effector

It is well known that no closed-form solutions can be found for the governing equations of object deformations. In this paper, we use a perturbation technique to derive an approximate solution. The solution of the perturbation technique is analytic instead of numerical. The latter may be obtained by the finite element or finite difference method. An analytic solution, however, is more convenient to analyze the deformation behavior than numerical solutions, and the computation for determining the deformation is simpler as well.

Deformable objects may be one, two or three dimensional. In our study, we only consider the first two since they are more frequently used in the manufacturing industry (such as composite material and metal sheet) and more tractable than three-dimensional objects. In the rest of this paper, one-dimensional object is called a beam and two-dimensional object is called a plate.

Consider a deformable object grasped by an end-effector (Fig. 2). A coordinate frame is attached to the beam with axis  $x$  along the length of the beam, axis  $y$  along the width, axis  $z$  along the thickness and the origin  $o$  at the grasped end of the beam. When the beam is horizontal, the coordinate frame  $(x, y, z)$  coincides with the world coordinate frame  $(X, Y, Z)$ . Now the problems becomes how much the deformations are as the orientation of the end-effector alters, and at what angle the deformation reaches the maximum. These problems need to be solved before we can identify a proper virtual space.

In considering the deformation of beam or plate grasping by an end-effector, the following two cases are possible and need to be studied:

- 1) Deformations of the beam as the end-effector pitches (Fig. 3), and
- 2) Deformations of the plate as the end-effector rolls (Fig. 4).

Once the deformation behavior of an object is understood, the virtual space problem can be solved.

This paper is organized as follows. Section 2 contains the deformation analysis of beams as the end-effector pitches. Section 3 analyzes deformation of plates as the end-effector rolls. Based on the results of the above two sections, we can predict deformation of beams or plates grasped by robot end-effectors that alter their orientations. Section 4 discusses the method for determining the virtual space. Finally the paper is concluded by Section 5.

## 2. DEFORMATION ANALYSIS OF THE BEAM AS THE END-EFFECTOR PITCHES

To determine the virtual space of the beam, one needs to know the degree of deformation as the pitching angle of the end-effector,  $\alpha$ , changes from  $0$  to  $\frac{\pi}{2}$ . According to Appendix, the governing equation of beam may be given by

$$D w_{,xxxx} = q + N_x w_{,xx} \tag{1}$$

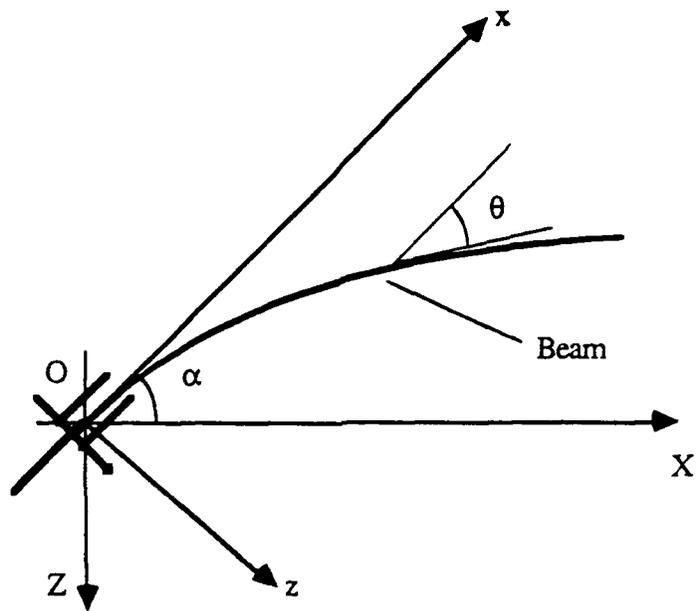


Fig. 3 Deformation of the beam as the end-effector pitches

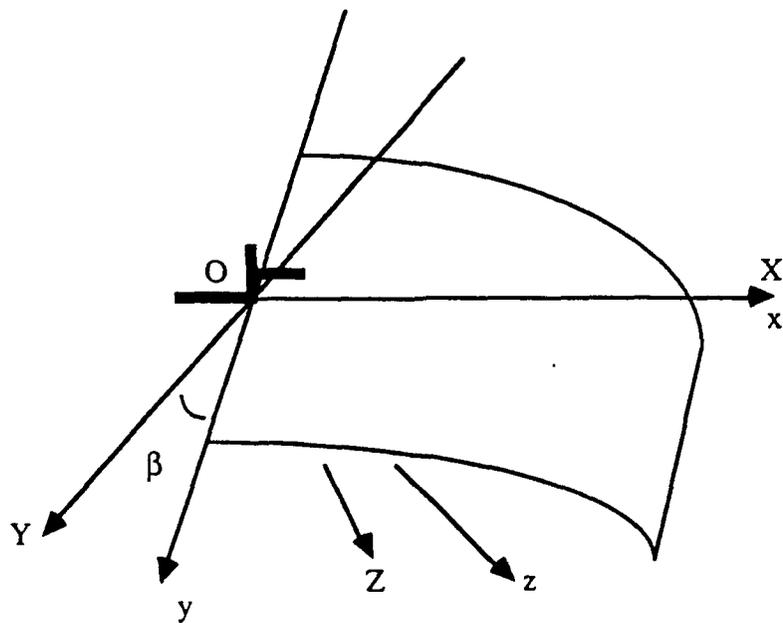


Fig. 4 Deformation of the plate as the end-effector rolls

where  $w(x,y)$  is the vertical displacement,  $D$  represents  $\frac{EI}{c}$  in which  $E$  is the modulus of elasticity,  $I$  is the inertia moment of the cross-section of the beam and  $c$  is the width of beam,  $q$  is the lateral load,  $N_x$  is the normal force with respect to the cross-section of the beam, and a subscript preceded by a comma denotes partial differentiation with respect to that subscript, i.e.  $w_{,xxxx} = \frac{\partial^4 w}{\partial x^4}$ , etc.

The above equation is obtained assuming that the coordinate system attached to the end-effector  $(x, y, z)$  coincides with the world coordinate system  $(X, Y, Z)$ . If the end-effector is turned by an angle, which amounts to turning the beam around axis  $Y$  (pitching), to angle  $\alpha$  (Fig. 3), the governing equation (1) has to be altered. However, if the deformation is considered with respect to the coordinates  $(x, y, z)$ , the governing equation will be in the same form as (1). That is,

$$D w_{,xxxx} = q_{\alpha} + N_{\alpha} w_{,xx} \quad (2)$$

where  $w(x)$  is the vertical displacement in the coordinate system  $(x, z)$ .  $N_{\alpha}$  is the normal force with respect to the cross-section of the beam which will be determined later, and  $q_{\alpha}$  is the lateral load in the coordinate system  $(x, o, z)$  and  $q_{\alpha} = q \cos \alpha$ .

To solve (2),  $N_{\alpha}$  must be known. This can be obtained as follows. In Fig. 5,  $N_{\alpha}$  may be written as

$$N_{\alpha} = q (a-x) \sin(\alpha-\theta) \quad (3)$$

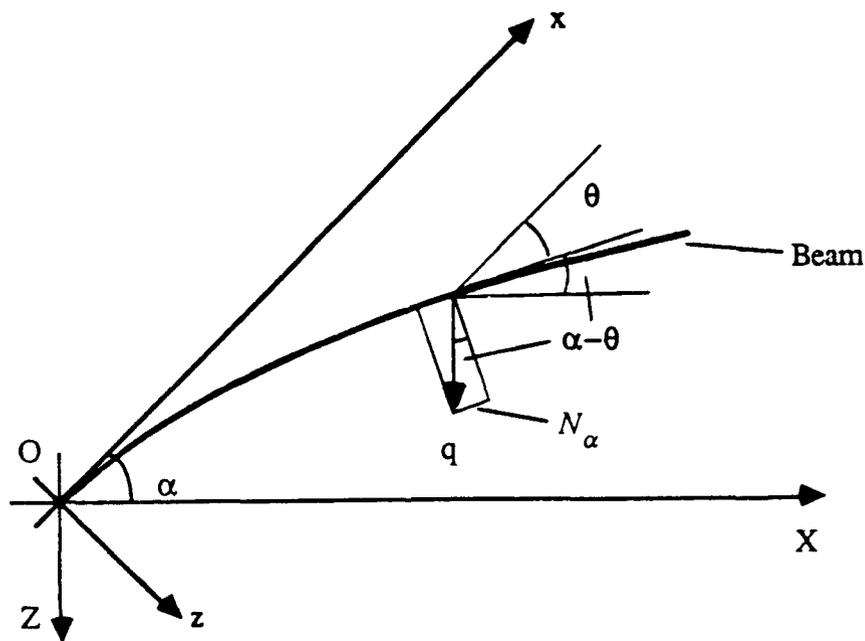


Fig. 5 Analysis of the in-plane force  $N_{\alpha}$  of the beam

where  $a$  is the length of the beam,  $\theta$  is the angle of slope,  $\tan \theta = w_x$ , and  $\theta=0$ , at  $x=0$ . In (3),  $q(a-x)$  is the force which acts at a point  $x$ , and caused by the lateral load along the length of  $(a-x)$ . Equation (3) expresses in-plane component of the force  $q(a-x)$ . Obviously, (3) satisfies the boundary condition (A-22) as defined in the Appendix.

Substituting (3) into (2) and dividing the result by  $D$  yields

$$w_{xxxx} = \frac{q}{D} \cos \alpha + \frac{q}{D} (a-x) \sin(\alpha-\theta) w_{xx} \quad (4)$$

If no external load is added to the beam, one has  $\frac{q}{D} = \frac{p}{EI}$  for the beam, where  $p$  is the load per unit length of the beam. As a result, (4) can be written as

$$w_{xxxx} = \frac{p}{EI} \cos \alpha + \frac{p}{EI} (a-x) \sin(\alpha-\theta) w_{xx} \quad (5)$$

In order to use the perturbation technique [10], we choose  $\epsilon = \frac{p}{EI}$  which is a small parameter since  $E$  is very large, and assume that  $w(x)$  can be developed into an ascending perturbation series of  $\epsilon$ . That is,

$$w(x) = \sum_{n=0}^{\infty} w_n(x) \epsilon^n \quad (6)$$

Substituting (6) into (5) and equating the terms of two sides which have the same order of the powers of  $\epsilon$  lead to a series of differential equations:

$$w_{0xxxx} = \frac{p}{EI} \cos \alpha \quad (7)$$

$$w_{1xxxx} = (a-x) (\sin \alpha - \cos \alpha w_{0x} + \dots) w_{0xx} \quad (8)$$

.....,

and the boundary conditions (see Appendix) become

$$w_n(0) = w_n(0)_x = w_n(a)_{xx} = w_n(a)_{xxx} = 0, \quad n=0, 1, \dots, \infty. \quad (9)$$

Solving (7) and (8) in conjunction with the boundary conditions of (9) gives

$$\begin{aligned}
w(x) = & \left( \frac{1}{24} x^4 - \frac{1}{6} a x^3 + \frac{1}{4} a^2 x^2 \right) \frac{P}{EI} \cos \alpha \\
& + \frac{1}{1680} \left( (a-x)^7 + 7 a^6 x - a^7 \right) \left( \frac{P}{EI} \right)^2 \sin \alpha \cos \alpha \\
& + \frac{1}{8} \left( \frac{1}{7560} x^{10} - \frac{1}{756} a x^9 + \frac{1}{168} a^2 x^8 - \frac{1}{1260} a^3 x^7 + \frac{1}{45} a^4 x^6 - \frac{1}{60} a^5 x^5 \right. \\
& \left. + \frac{1}{84} a^7 x^3 - \frac{3}{280} a^8 x^2 \right) \left( \frac{P}{EI} \right)^3 \cos^3 \alpha + \dots
\end{aligned} \tag{10}$$

In [11] it is mentioned that in the equations of large deflections,  $x$  represents the length of the arc. The results of (10), therefore, present the relationship between the deflections and the arc length of the beam.

With  $x=a$ , (10) becomes (we truncate the series of (10) and just choose the first three terms for simplicity):

$$w(a) = \frac{1}{8} a^4 \frac{P}{EI} \cos \alpha + \frac{1}{280} a^7 \left( \frac{P}{EI} \right)^2 \sin \alpha \cos \alpha - \frac{1}{2240} a^{10} \left( \frac{P}{EI} \right)^3 \cos^3 \alpha \tag{11}$$

When  $\alpha=0$ , i.e. the pitch angle of the end-effector is zero, the second term of (11) becomes zero. It is found that the first term of (11) is in agreement with the small deflection theory and the third, which represents large deflections of the beam, agrees to the results of reference [8] ( pp. 173).

When  $\alpha \neq 0$ , the second term of (11), which represents the contribution of the in-plane forces to the beam deflection, will be in effect. We may differentiate (11) with respect to  $\alpha$  and obtain the maximum deflection,  $w_{max}$ . For example, when  $a=4.0$  and  $\frac{P}{EI}=0.25$ , one may find that the maximum deflection occurs when  $\alpha=1.02$ , which is  $w_{max}=1.50742$ . It should be noted that the maximum deflection does not take place when the pitch angle is zero.

Based on the above results we may construct the virtual space of the beam. This topic will be discussed in Section 4.

### 3. DEFORMATION ANALYSIS OF THE PLATE AS THE END-EFFECTOR ROLLS

Consider a rectangular plate grasped by an end-effector (Fig. 4) which is deformed by its own weight. Deformation behavior of cantilever plates was previously studied by many works such as

[12-15]. But it has never been studied when a elastic plate grasped by an end-effector and the end-effector rolls. In this section, we present a solution for this particular case. For simplicity, we assume that the edge of the plate which is grasped by the end-effector has no deformation. This assumption agrees to many practical cases, since when the edge is grasped by the end-effector, the effective length of deformation is reduced by two. This, in practice, makes the deformation of the edge ignorable if the edge is not too long. Once again, the perturbation technique will be used to find the approximate solution.

In the coordinate system  $(x, y, z)$ , the governing equation for the plate may be written as (see (A-2) in the Appendix)

$$w'_{xxxx} + 2 w'_{xxyy} + w'_{yyyy} = \frac{q}{D} \cos\beta + \frac{1}{D} ( N_{\beta y} w'_{yy} + N_{\beta x} w'_{xx} + 2 N_{\beta xy} w'_{xy} ) \quad (12)$$

where  $N_{\beta y}$ ,  $N_{\beta x}$ , and  $N_{\beta xy}$  are in-plane forces.

It is noted that  $N_{\beta xy}$  is the shear force which is summation of the shear stresses in the thickness direction of the plate. The shear force  $N_{\beta xy}$  has little contribution to the vertical deflections of the plate, and can be considered null ( $N_{\beta xy} = 0$ ). In addition, a similar force analysis for  $N_{\beta x}$  can be made as shown in Fig. 5, i.e.,  $N_{\beta x}$  may be written as

$$N_{\beta x} = q (a-x) \cos\beta \sin(-\theta_1) \quad (13)$$

where  $a$  is the length of the rectangular plate, and  $\theta_1$  is the slope angle along axis  $x$ , and  $tg\theta_1 = w'_{,x}$ . In this section, we consider that the end-effector has a zero pitching angle, i.e.,  $\alpha=0$ , but rolls by an angle  $\beta$ . In fact, if one lets  $\alpha$  in (3) be 0 and  $q\cos\beta$  be in the place of  $q$ , (3) becomes (13).

It is clear that the force  $N_{\beta y}$  is a function of variable  $y$ , and the other component of  $q$ ,  $q\sin\beta$ , should be included in the force  $N_{\beta y}$ . In addition,  $N_{\beta y}$  has to satisfy the boundary conditions shown later in (26). Therefore, the expression of  $N_{\beta y}$  should include factor  $(y^2 - b^2)$ , where  $b$  is the half-width of the plate. Furthermore, when the end-effector rolls to angle  $\beta$ , the deformations along axis  $y$  may be described by angle  $\theta_2$ , which is the slope angle along axis  $y$  and a function of  $x$  and  $y$ ; and  $tg\theta_2 = w'_{,y}$ . It can be seen that the force  $N_{\beta y}$  will equate to zero when

$$\beta + \theta_2 = 0 \quad (14)$$

Solving (14) for  $y$ , one may obtain a function as follows

$$y^* = f(x, \beta). \quad (15)$$

The curve governed by  $y^* = f(x, \beta)$  is called Zero- $N_{\beta y}$  curve. Along the Zero- $N_{\beta y}$  curve of the plate,  $N_{\beta y}$  is equal to zero. Therefore, the expression of  $N_{\beta y}$  should include factor  $(y - y^*)$ . For simplicity, we let

$$y^* = f(x, \beta) = 0 \quad (16)$$

which means the Zero- $N_{\beta y}$  curve is axis  $x$ . To summarize the above,  $N_{\beta y}$  may be approximately expressed as

$$N_{\beta y} = q \sin\beta (y^2 - b^2) (y - y^*) / k \quad (17)$$

where  $k$  is a constant. The question now becomes: how to determine  $k$ ?

When  $y$  is equal to  $\pm \frac{1}{2}b$ ,  $N_{\beta y}$  should equate to  $\pm \frac{1}{2}b q \sin\beta$  (here we let  $y^* = 0$ ). By substituting the above two values of  $y$  and  $N_{\beta y}$  into (17), one may obtain  $k = \frac{3}{4}b^2$ . Once  $k$  is given, the relation between the force  $N_{\beta y}$  and  $y$  can be plotted as shown in Fig. 6.

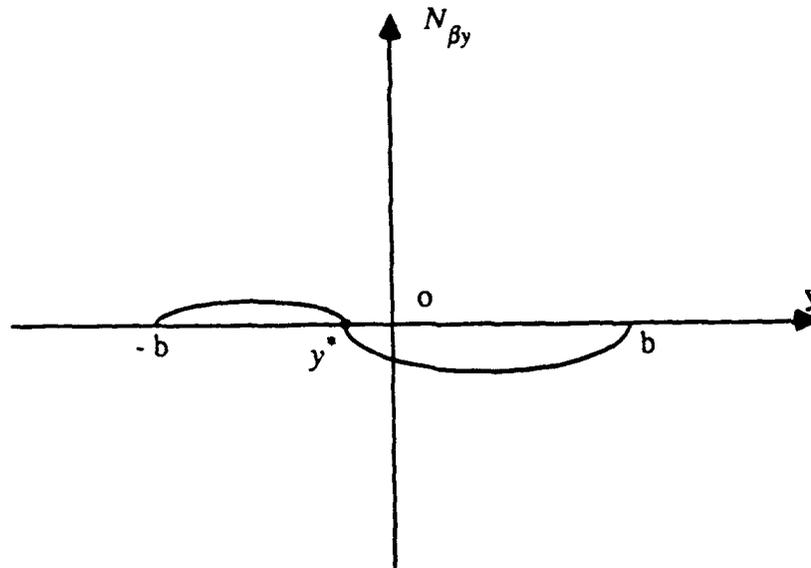


Fig. 6 Distribution of the force  $N_{\beta y}$  in a section  $y$ .

Now (12) may be written as

$$w_{,xxxx} + 2 w_{,xxyy} + w_{,yyyy} = \frac{q}{D} \cos\beta + \frac{q}{D} \sin\beta (y^2 - b^2) y w_{,yy} / \left(\frac{3}{4} b^2\right) + (a-x) \cos\beta \sin(-\theta_1) w_{,xx} \quad (18)$$

and the boundary conditions are as follows:

a. for the edge grasped by the end-effector:

$$w = 0, \quad \text{at } x=0 \quad (19)$$

$$w_{,x} = 0, \quad \text{at } x=0 \quad (20)$$

b. for the free edge [11]:

$$w_{,xx} + \mu w_{,yy} = 0, \quad \text{at } x=a \quad (21)$$

$$w_{,xxx} + (2-\mu) w_{,yyx} = 0, \quad \text{at } x=a \quad (22)$$

$$w_{,yy} + \mu w_{,xx} = 0, \quad \text{at } y= \pm b \quad (23)$$

$$w_{,yyy} + (2-\mu) w_{,xyy} = 0, \quad \text{at } y= \pm b \quad (24)$$

$$N_{\beta x} = 0, \quad \text{at } x=a \quad (25)$$

$$N_{\beta y} = 0, \quad \text{at } y= \pm b \quad (26)$$

where  $\mu$  is the Poisson's ratio. Clearly, (13) satisfies the boundary condition (25).

By using the perturbation technique, the solution of (18) is expressed by

$$w(x,y) = \sum_{n=0}^{\infty} w_n(x,y) \varepsilon^n \quad (27)$$

where  $\varepsilon = \frac{q}{D}$ . Substituting (27) into (18) and equating the terms of the two sides which have the same powers of  $\varepsilon$ , we can obtain a series of differential equations:

$$w_{0'xxxx} + 2 w_{0'xxyy} + w_{0'yyyy} = \frac{q}{D} \cos\beta \quad (28)$$

$$w_{1'xxxx} + 2 w_{1'xxyy} + w_{1'yyyy} = \sin\beta (y^2 - b^2) y w_{0'yy} / \left(\frac{3}{4} b^2\right) \\ + (a-x) \cos\beta \sin(-\theta_1) w_{0'xx} \quad (29)$$

.....

To solve (28), we assume that the solution may be expressed as:

$$w_0(x,y) = x^2 ( A_1 + A_2 x + A_3 x^2 + A_4 y^2 + A_5 x^3 + A_6 xy^2 \\ + A_7 x^4 + A_8 x^2 y^2 + A_9 y^4 ). \quad (30)$$

Substituting (30) into (28) and equating the constant term to  $\frac{q}{D} \cos\beta$  and the other terms to zero, we can get a series of algebraic equations. Note that the boundary conditions of (19) and (20) are automatically satisfied by (30). By using (30) and boundary conditions (21) - (24), another set of algebra equations can be established. By solving the two sets of algebra equations, all the  $A_i$  can be solved.

In order to solve (29), the solution of (29) is assumed to be

$$w_1(x,y) = x^2 ( B_1 + B_2 x + B_3 y + B_4 x^2 + B_5 xy + B_6 y^2 + B_7 x^3 + B_8 x^2 y + B_9 xy^2 + B_{10} y^3 \\ + B_{11} x^4 + B_{12} x^3 y + B_{13} x^2 y^2 + B_{14} xy^3 + B_{15} y^4 + B_{16} x^4 y + B_{17} x^3 y^2 + B_{18} x^2 y^3 \\ + B_{19} xy^4 + B_{20} y^5 + B_{21} x^4 y^2 + B_{22} x^3 y^3 + B_{23} x^2 y^4 + B_{24} xy^5 + B_{25} y^6 + B_{26} x^4 y^3 \\ + B_{27} x^3 y^4 + B_{28} x^2 y^5 ) \quad (31)$$

More terms may be included in (31) to improve the accuracy of the solution (the same is true for (30)). By using the same approach as used for solving  $A_i$ , all the  $B_i$  can be solved.

After  $A_i, B_i$  are obtained, the relationship between the deflection  $w(x,y)$  and the variables  $x$  and  $y$  is established. As a result, we can calculate the maximal deflection of the plate and construct

its virtual space. As an example, if the parameters of the plate are chosen as  $a=4.0$ ,  $b=1.0$ ,  $\mu=0.25$ , and  $\frac{q}{D}=0.25$ , the maximum deflection in the direction of axis  $z$ ,  $w_{max}$ , is calculated to be 1.67339 at  $\beta = 0.49$ , and occurs at the corner of the plate where  $x=4.0$ ,  $y=-1.0$ .

#### 4. DETERMINE THE VIRTUAL SPACE OF DEFORMABLE OBJECTS

In Sections 2 and 3, we derived expressions of deformation for the beam and plate, respectively. According to the expression, the maximum deflections of a beam or a plate can be obtained. From the maximum deflection, one can determine the virtual space of the object, and further treat the object as a rigid object. By using the virtual space approach, the handling of the deformable object becomes a simple problem.

For practical purpose, we propose to use the following two criteria to select an optimal virtual space:

**Criterion 1:** The virtual space should have a smallest volume, but cover all the deformed shapes of the elastic object;

**Criterion 2:** The virtual space should be simple to compute.

A good selection of the virtual space might have to balance these two criteria which are trade-offs to each other. In the following, we will present a few possible virtual spaces for the beam and plate deformations.

##### A. The Virtual Space of the Beam

By studying the deformation behavior of the beam, we propose two geometrical shapes as the virtual space. They are rectangle and triangle respectively. The formulation for each virtual space is discussed as follows.

##### A.1 Rectangle virtual space

When the object is a beam, deformation occurs in the  $(x,z)$  coordinates. The virtual space in this case is a rectangle which enclose the maximum deformation of the object. The lines that bounds the rectangle can be described by the following equations in the end-effector coordinates:

$$x=0, x=a, z=0, \text{ and } z=w_{max} \quad (32)$$

where  $w_{max}$  is the maximum deflection of the beam (see Fig. 7 for the shape of the rectangular virtual space). If the object is a plate, the virtual space should have the third dimension which is just a linear extension of the rectangle shape in the  $y$  direction. In this case, equation (32) describes the surfaces bounding the virtual space. In addition, the virtual space will be bounded by another two surfaces:  $y = \frac{c}{2}$  and  $y = -\frac{c}{2}$  where  $c$  is the width of the plate which is usually much shorter than the length of the plate and has no deformation.

This kind of virtual space can cover all deformations of the beam. And the equation of the virtual space do not change as the end-effector pitches, i.e., the virtual space is fixed to the end-effector. The virtual space clearly meets the second criterion since the rectangle has a simple geometric shape. However, the volume of the space is not optimal. This can be easily shown by a smaller virtual space of the following.

### A.2 Triangle virtual space

This kind of virtual space uses a triangular shape as shown in Fig. 8. The three lines that bounds the triangle has the following expression:

$$x=a, z=0, \text{ and } z=w_{max} x/x_d \quad (33)$$

where  $w_{max}$  is the maximum deflection, and  $x_d$  is the  $x$  coordinate of the maximum deflection point. It is clear that the triangle has a smaller area than the rectangle. Therefore, this virtual space can meet the two criteria better than the rectangular virtual space. If the object is a plate, the virtual space becomes a prism. The above equations represent surfaces that bound the prism. In addition, two more surfaces that bounds the prism can be described as  $y = \frac{c}{2}$  and  $y = -\frac{c}{2}$  respectively, where  $c$  is the width of the plate.

The virtual space for a beam deformation can be further refined if more computation is involved. For example, if the shape of the maximum deformation can be exactly described as  $z=f(x)$ , a more optimal virtual space is bounded by  $x=a, z=0$  and  $z=f(x)$ .

### B. Virtual Space of the Plate

The two kinds of virtual space as discussed above may be extended to the case of plate. If the width of the plate is denoted as  $2b$  instead of  $c$ , one may have the rectangle type of virtual space be described as

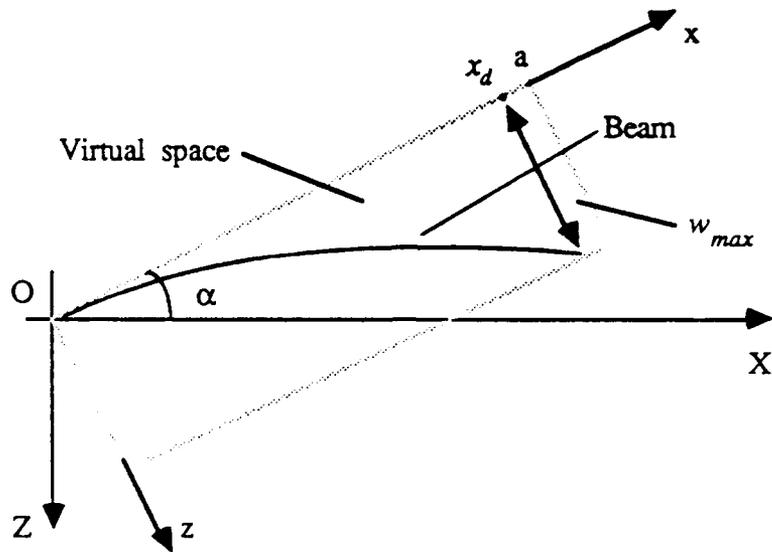


Fig. 7 The first kind of virtual space of the beam

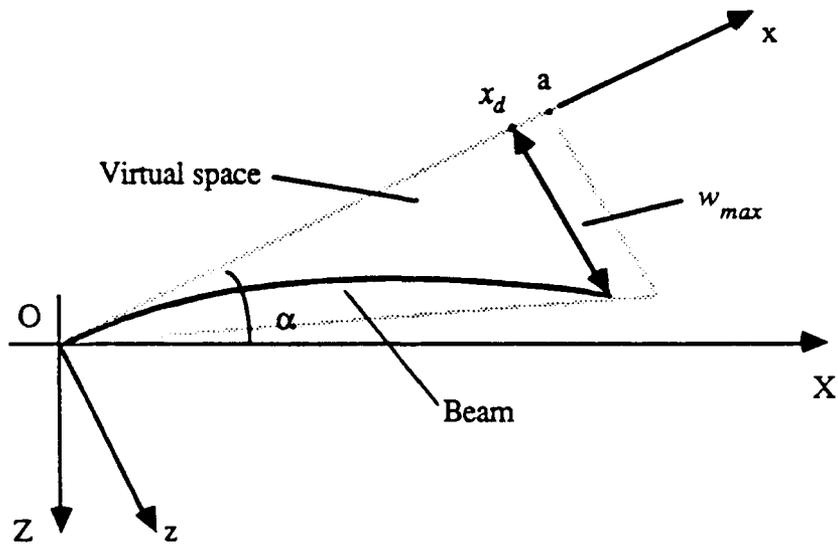


Fig. 8 The second kind of virtual space of the beam

$$x=0, x=a, y=b, y=-b, z=0, \text{ and } z=w_{max} \quad (34)$$

and the triangle type of virtual space be described as

$$x=0, x=a, y=b, y=-b, z=0, \text{ and } z=w_{max} x/x_d \quad (35)$$

where  $w_{max}$  is the maximum deflection of the plate and is calculated by using (27), (28) and (29).

In addition to the above two types, a new type of virtual space is also possible for the plate deformation. This can be described by the following equations:

$$\begin{aligned} x=0, x=a, y=b, y=-b, z=0, z=w_{1max} x/x_d \text{ ( at } y=-b) \text{ and} \\ z=w_{2max} x/x_d \text{ ( at } y=b) \end{aligned} \quad (36)$$

where  $w_{1max}$  is  $w_{max}$ , and  $w_{2max}$  is the deflection of another corner ( $x=x_d, y=b$ ) of the plate, as shown in Fig. 9. This type of virtual space become less regular than the first two types, but has a smaller volume than the first two. For example, for the size and deformation of the plate as defined at the end of the previous section, the volume of the space is  $4.0 \times 2.0 \times 1.67339$  when the rectangular type is employed, become half of that if the triangular type is used, and even smaller when the irregular type is used.

## 5. CONCLUSIONS

Path planning for deformable objects is a difficult problem because of uncertainty of the object shape. In this paper, we propose a virtual space approach to solve the problem. The virtual space is a hypothetical space which covers all the possible shapes of the deformable object. As a result, the object can be treated as a rigid body, and the path planning problem becomes simple to solve. In summary, the following contributions have been made by this paper:

- a. Solutions to beam and plate deformation are derived when the objects are handled by a robot end-effector with different orientations which include both pitching and rolling.
- b. The criteria for constructing optimal virtual spaces are proposed. Based on the criteria, several methods for determining the virtual space of beams and plates are suggested.

The determination of the virtual space, however, remains to be an open problem when an object is deformed in a more complicated shape or even smaller virtual space needs to be found. These problems will be investigated in our future research.

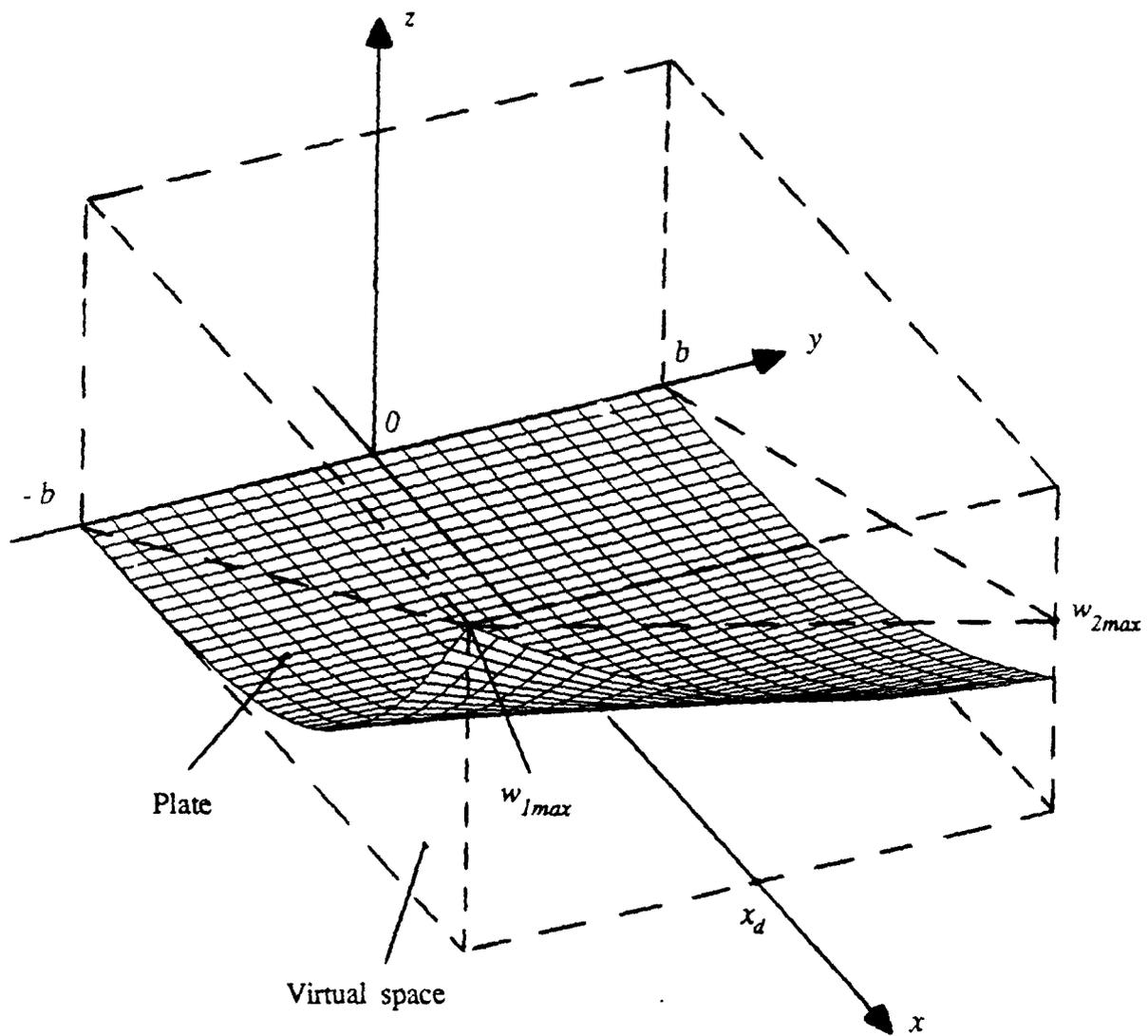


Fig. 9 Deformation of the plate at  $\beta=0.49$  and its virtual space

## APPENDIX

The governing equations of deformation of a rectangular plate (Fig.1), under large deflections, can be expressed as follows ( von Karman equations ) [11]:

$$F_{,xxxx} + 2 F_{,xxyy} + F_{,yyyy} = E ( ( w_{,xy} )^2 - w_{,xx} w_{,yy} ) \quad (\text{A-1})$$

$$w_{,xxxx} + 2 w_{,xxyy} + w_{,yyyy} = \frac{q}{D} + \frac{1}{D} ( N_y w_{,yy} + N_x w_{,xx} + 2 N_{xy} w_{,xy} ) \quad (\text{A-2})$$

where previously undefined terms  $F$  is the stress function,  $N_x$  is the normal force with respect to section  $y$ ,  $N_y$  is the normal force with respect to section  $x$ , and  $N_{xy}$  is the shear force.

According to [11], the in-plane forces  $N_x$ ,  $N_y$  and  $N_{xy}$  can be expressed by the stress function  $F$ . That is

$$N_x = h F_{,yy} \quad (\text{A-3})$$

$$N_y = h F_{,xx} \quad (\text{A-4})$$

$$N_{xy} = - h F_{,xy} \quad (\text{A-5})$$

where  $h$  is the thickness of the plate.

The boundary conditions of the plate can be defined as (Fig. 2):

a. for the edge clamped by the end-effector:

$$w = 0, \quad \text{at } x=0 \quad (\text{A-6})$$

$$w_{,x} = 0, \quad \text{at } x=0 \quad (\text{A-7})$$

b. for the free edge:

$$M_x = 0, \quad \text{at } x=a \quad (\text{A-8})$$

$$V_x + M_{xy'y} = 0, \quad \text{at } x=a \quad (\text{A-9})$$

$$N_x = N_y = N_{xy} = 0, \quad \text{at } x=a \quad (\text{A-10})$$

where  $M_x$  is the bending moment about axis  $y$ ,  $M_{xy}$  is the torsional moment,  $V_x$  is the shear force with respect to section  $y$ , and  $a$  is the length of the rectangular plate.

Now we analyze a particular case, i.e., the object is a beam. Consider a beam grasped by an end-effector. It is obvious that the deflection along axis  $y$  may be neglected. That is, the vertical displacement  $w(x,y)$  turns to be  $w(x)$ . It follows that

$$w'_{yyyy} = w'_{xxyy} = w'_{xyy} = w'_{yy} = w'_{xy} = 0 \quad (\text{A-11})$$

Thus (A-1) becomes trivial, and (A-2) becomes

$$D w_{xxxx} = q + N_x w_{xx} \quad (\text{A-12})$$

Now consider the boundary conditions of the beam. According to [11], we have the following expression:

$$M_x = -D ( w_{xx} + \mu w_{yy} ) \quad (\text{A-13})$$

where  $M_x$  is the bending moment about axis  $y$ , and  $\mu$  is the Poisson's ratio. Using (A-11) and (A-13), (A-8) becomes

$$w_{xx} = 0, \quad \text{at } x=a \quad (\text{A-14})$$

where  $a$  is the length of the beam. In addition, we have

$$M_{xy} = 0, \quad \text{at } x=a \quad (\text{A-15})$$

where  $M_{xy}$  is the torsional moment, and the expression [11]:

$$V_x = -D ( w_{xxx} + w_{xyy} ) \quad (\text{A-16})$$

where  $V_x$  is the shear force with respect to section  $y$ . Consequently, (A-9) becomes

$$w_{xxx} = 0, \quad \text{at } x=a \quad (\text{A-17})$$

In summary, the boundary conditions of the beam can be listed as follows:

- a. for the edge clamped by the end-effector:

$$w = 0, \quad \text{at } x=0 \quad (\text{A-18})$$

$$w'_{,x} = 0, \quad \text{at } x=0 \quad (\text{A-19})$$

b. for the free edge:

$$w'_{,xx} = 0, \quad \text{at } x=a \quad (\text{A-20})$$

$$w'_{,xxx} = 0, \quad \text{at } x=a \quad (\text{A-21})$$

$$N_x = 0, \quad \text{at } x=a. \quad (\text{A-22})$$

## REFERENCES

1. R. P. Singh, R. J. van der Voort, and P. W. Likins, "Dynamics of flexible bodies in tree topology -- A computer oriented approach", *Journal of Guidance, Control and Dynamics*, vol. 8, no. 5, pp. 584 -590, 1985.
2. T. R. Kane, R. R. Ryan, and A. K. Banerjee, "Dynamics of a beam attached to a moving base", *ibid.* vol. 10, no. 2, pp. 139-151, 1987.
3. A. K. Banerjee and T. R. Kane, "Dynamics of a plate in large overall motion", *Trans. ASME Journal of Applied Mechanics*, vol. 56, pp. 887-892, December, 1989
4. A. Villarreal and H. Asada, "A geometric representation of distributed compliance for the assembly of flexible parts", *Proc. 1991 IEEE International Conference on Robotics and Automation*, Sacramento, CA, April 9-11, 1991, pp. 2708-2715.
5. Y. F. Zheng, R. Pei and C. Chen, "Strategies for automatic assembly of deformable objects", *Proc. 1991 IEEE International Conference on Robotics and Automation*, Sacramento, CA, April 9-11, 1991, pp. 2598-2603.
6. M. Z. Chen and Y. F. Zheng, "Automated handling of deformable objects by coordinated manipulators", to appear .
7. C.-Y. Chia, Nonlinear Analysis of Plates, McGraw-Hill, Inc. 1980.
8. R. Frisch-Fay, Flexible Bars, Butterworth & Co. Limited, 1962.
9. M. Z. Chen and Y. F. Zheng, "Dynamic behavior of the beam handled by a robot end-effector", to appear.
10. A. H. Nayfeh, Introduction to Perturbation Techniques, John Wiley & Sons, Inc. 1981.
11. C. P. Heins, Applied Plate Theory for the Engineer, D.C. Heath and Company, 1976.
12. W. A. Nash, "Several approximate analyses of the bending of a rectangular cantilever plate by uniform normal pressure", *Trans. ASME Journal of Applied Mechanics*, vol. 19, no. 1, pp. 33-36, 1952.
13. H. J. Plass, Jr., J. H. Gaines and C. D. Newson, "Application of Reissner's variational principle to cantilever plate deflection and vibration problems", *Trans. ASME Journal of Applied Mechanics*, vol. 29, no. 1, pp. 127-135, 1962.
14. T. H. Lin, S. R. Lin and B. Mazelsky, "Large inextensional deflection of thin cantilevered plates", *Trans. ASME Journal of Applied Mechanics*, vol. 35, no. 4, pp. 774-777, 1968.
15. P. Darmon and R. C. Benson, "Large inextensional deformation of orthotropic cantilevered plates with distributed loads", *Trans. ASME Journal of Applied Mechanics*, vol. 52, no. 2, pp. 385-388, 1985.